Signal processing techniques for GNSS anti-spoofing algorithms

ScuDo

Scuola di Dottorato ~ Doctoral School

WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation

Doctoral Program in Electronics Engineering (29^{th}cycle)

Signal processing techniques for GNSS anti-spoofing algorithms

By

Esteban Garbin Manfredini

******

Supervisor(s):

Prof. Fabio Dovis, Supervisor

Doctoral Examination Committee:

Prof. Ali Broumandan , Referee, University of Calgary

Prof. Heidi Kuusniemi, Referee, Finnish Geospatial Research Institute

Politecnico di Torino

2017

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Esteban Garbin Manfredini
2017

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

I’d take the awe of understanding over the awe of ignorance any day

† *Douglas Adams*

...to my parents, my sister and Marta

Acknowledgements

I would like to greatly thank my supervisor prof. Fabio Dovis, for the patience and guidance that he provided through the whole PhD. process. For keeping up with my mistakes and guiding me to the right path in each of the topics discussed throughout the PhD, and for giving me the opportunity to travel abroad and meet so many people and learn so many things.

Many thanks to Beatrice Motella, for the help with the design of the multidimensional ratio test and the development of the statistical description.

Thanks to the Bomber, for the collaboration done together, for all the interesting discussions we had, for being a good travel partner and for all the laughs and fun moments.

My most sincere thanks to everyone in the lab, especially to Rodrigo, Luciano, Nicola, Calogero, Marco and Falco for all the help through the period of the PhD. studies, all the events we shared, both in academics and non academics environments and in general, the fun times we shared. Thanks to all other members of the NavSas group that I have met, for the welcoming environment and the availability to help whenever asked.

I would like to thanks prof. Dennis M. Akos, who supervised my stay in Colorado University, for such a warming welcoming during my period at Boulder, for the guidance and help provided and for enabling such an amazing period of time. I'm very thankful to everyone I met in prof. Akos lab, for welcoming us as if we had been there for a long time and for sharing all the travels, gatherings and always having a good time. You guys made the stay at Boulder an even more enriching experience. Thanks particularly to Damian for his stories of the Russian Imperial Stout.

I would like to extend my gratitude to the researchers at Stanford GPS lab, for the guidance and help with the proposed anti-spoofing technique and to Zeta associates for the help with the provided data collections.

Gracias a Marta, porque me ha acompanado en esta aventura a pesar de que fuera un camino tortuoso y dificil a veces. Gracias por estar ahi desde el principio y por ir al otro lado del mundo conmigo. Gracias por siempre alentarme y apoyarme para seguir adelante con este proceso, eres la luz que ilumina mis dias y no lo hubiera logrado sin ti.

Muchas gracias a papa y mama, porque sin ustedes no hubiera podido hacer nada de esto. Gracias por su apoyo incondicional y por estar siempre presentes a pesar de la distancia. Gracias a mi hermana por ser una verdadera amiga y saber que siempre estaremos unidos no importa la distancia. Gracias por la gioia di essere insieme.

Alla fine, volevo ringraziare a tutti gli amici di Torino che hanno dovuto sopportare la mia presenza per questi tre anni e che tante volte mi hanno offerto un passaggio, un piatto di pasta o un posto per dormire e mi hanno sempre accolto tra di loro. Grazie mille per tutti i giochi.

Thanks to everyone I met during these three years of PhD studies. Thanks to all the professors, researchers, classmates, students and friends for the knowledge you have shared.

Abstract

The Global Navigation Satellite Systems (GNSS) usage is growing at a very high rate, and more applications are relying on GNSS for correct functioning. With the introduction of new GNSSs, like the European Galileo and the Chinese Beidou, in addition to the existing ones, the United States Global Positioning System (GPS) and the Russian GLONASS, the applications, accuracy of the position and usage of the signals are increasing by the day.

Given that GNSS signals are received with very low power, they are prone to interference events that may reduce the usage or decrease the accuracy. From these interference, the spoofing attack is the one that has drawn major concerns in the GNSS community. A spoofing attack consist on the transmission of GNSS-like signals, with the goal of taking control of the receiver and make it compute an erroneous position and time solution.

In the thesis, we focus on the design and validation of different signal processing techniques, that aim at detection and mitigation of the spoofing attack effects. These are standalone techniques, working at the receiver's level and providing discrimination of spoofing events without the need of external hardware or communication links. Four different techniques are explored, each of them with its unique sets of advantages and disadvantages, and a unique approach to spoofing detection. For these techniques, a spoofing detection algorithm is designed and implemented, and its capabilities are validated by means of a set of datasets containing spoofing signals. The thesis focuses on two different aspects of the techniques, divided as per detection and mitigation capabilities. Both detection techniques are complementary, their joint use is explored and experimental results are shown that demonstrate the advantages.

In addition, each mitigation technique is analyzed separately as they require specialized receiver architecture in order to achieve spoofing detection and mitigation. These techniques are able to decrease the effects of the spoofing attacks, to the point

of removing the spoofing signal from the receiver and compute navigation solutions that are not controlled by the spoofer and lead in more accurate end results.

The main contributions of this thesis are: the description of a multidimensional ratio metric test for distinction between spoofing and multipath effects; the introduction of a cross-check between automatic gain control measurements and the carrier to noise density ratio, for distinction between spoofing attacks and other interference events; the description of a novel signal processing method for detection and mitigation of spoofing effects, based on the use of linear regression algorithms; and the description of a spoofing detection algorithm based on a feedback tracking architecture.

Contents

List of Figures

List of Tables

List of Acronyms

ADC Analog to Digital converter

ALL Amplitude Lock Loop

AGC Automatic Gain Control

ARNS Aeronautical Radionavigation Service

AWGN Additive White Gaussian Noise

BDE Barycenter Delay Estimation

BPSK Binary phase-shift keying

C/N_0 Carrier to Noise density ratio

C/A Coarse/Acquisition

CAF Cross Ambiguity Function

CDMA Code Division Multiple Access

COTS Commercial off-the-shelf

DC Direct Current

DLL Delay Lock Loop

DW Detection Window

ECADLL Extended Coupled Delay Lock Loop

FDMA Frequency Division Multiple Access

FFT Fast Fourier Transform

FIR Finite Impulse Response

FLL Frequency Lock Loop

GEO Geostationary orbit

GNSS Global Navigation Satellite Systems

GPS Global Positioning System

HOW Hand Over word

IGSO Inclined geo-synchronous orbits

ICD Interface control document

IF Intermediate frequency

KF Kalman Filter

LAF Linear Adaptive Filter

LASSO Least Absolute Shrinkage and Selection Operator

LNA Low Noise Amplifier

LOS Line Of Sight

LR Likelihood Ratio

LS Least Squares

MEO Medium earth orbits

MPDD Multipath Distance Detector

NP Neyman-Pearson

PLL Phase Lock Loop

PPD Personal privacy device

PVT Position Velocity and Time

PRN Pseudo Random Noise

RAIM Receiver Autonomous Integrity Monitoring

RF Radio Frequency

RFI Radio Frequency Interference

RINEX Receiver Independent Exchange Format

RM Ratio Metric

RT Ratio Test

SIS Signal in Space

SoL Safety of life

SQI Signal Quality Index

SQM Signal Quality Monitoring

TEXBAT Texas Anti Spoofing Test Battery

TJ Time Jumper

TLM Telemetry word

TOA Time of Arrival

TOW Time of Week

USRP Universal Software Radio Peripheral

WAAS Wide Area Augmentation System

Introduction

As technological advances are introduced in society and their use spreads among the people, more and more applications are found for each technology. Global Navigation Satellite system (GNSS) technology is a clear example of this phenomenon. Ever since the Global Positioning System (GPS) became operational, its applications and use have increased dramatically. Nowadays, almost every person has a device with them, capable of guiding them through the ever-changing cities by means of GNSS signals. Additionally, these devices are supported by infrastructures that are synchronized thanks to these GNSS signals. Many other examples can be found to understand how ubiquitous GNSSs are in everyday activities.

Technology evolves and spreads, and the concerns for security in all electronic and telecommunication systems increase as well. This concern applies to many different sectors of today's society, one of them being GNSS. As can be seen, modern society strongly relies on GNSS, for a constantly increasing number of applications and services. However, the issues related to the security of such systems are sometimes underestimated. This is the case of some services relying on GNSS civil signals. In fact, the menace of intentional radio-frequency interference, such as jamming or spoofing attacks, is gaining momentum, and discussions are being held, trying to find ways to protect GNSS civil users from these attacks.

Nowadays, the effects of these intentional interferences, which are able to compromise the correct functioning of the GNSS receivers are well known [61, 76, 23, 30, 86], and the need for improving the security of the receiver has been demonstrated [68, 2], especially in case of applications whose malfunctioning would put people's safety at risk.

Among the different interference attacks that can affect GNSSs, one of the most dangerous is the spoofing attack. It consists of the transmission of GNSS-like signals,

aligned with the satellite signals, with the goal of taking control of the position velocity and time (PVT) solution that the receiver computes. In this way, the attacker is able to fake the target position without being noticed and may cause severe damage to the applications relying on the GNSS signal.

In this thesis, we discuss different anti-spoofing techniques and their performances against live spoofing attacks. The techniques presented here are based on signal processing algorithms that can be implemented in the receiver to discriminate spoofing presence. All of these have the common characteristic of being contained inside the GNSS receiver, without the need for external hardware, communications links or modifications of the Signal in Space (SIS). In addition, a subset of these techniques is based on detection of the spoofing signals while the other is capable of detection and mitigation of the spoofing effects.

Thesis Outline

The thesis is divided in three parts, with a total of seven Chapters. A brief description of each Chapter follows:

- **Part I: Basics of GNSS and Spoofing attacks**
  - **Chapter 1** introduces the basic concepts of GNSS. It gives a brief explanation on the basic concept of position, velocity and time (PVT), and how to obtain it from the signal transmitted by the satellite. A brief status review of the different GNSS systems is presented. It also presents the basic structure of the GNSS signals, highlighting aspects that will be used by the spoofing detection techniques and detailing the differences between signals from different services. Finally, the basic GNSS receiver structure is described and the basic functionality of each of its components are detailed.
  - **Chapter 2** describes the basic aspects of the spoofing threat. It shows the different interference effects that can disturb GNSS receivers and it describes the effects of each one. A classification of the different spoofing attack scenarios is presented, focusing on the classes that will be considered for the results shown in the thesis. A literature review

is provided for the different anti spoofing techniques that have been proposed. Finally, a brief discussion highlighting the difference between spoofing detection techniques and spoofing mitigation techniques is presented, in order to clarify the structure used for the thesis.

- **Part II: Spoofing Detection**
  - **Chapter 3** presents the basics of the Signal Quality Monitoring (SQM) technique. It describes two of the different tests that can be performed to detect anomalies in the correlation function. It presents the mathematical derivation of the detection theory for the ratio test, which will be used in the thesis. A description of the adopted detection algorithm and the rationale behind are presented. Finally, the results obtained by means of the SQM metric are presented, testing the technique against a dynamic and a static spoofing dataset.
  - **Chapter 4** presents a multidimensional ratio metric, that includes the ratio test metric, a temporal check, and observations on the number of satellites. The metric is capable to distinguish between spoofing attacks and the presence of multipath signals. The mathematical derivation of the statistical characteristics of the metric is presented. Finally, results for the spoofing detection capabilities of the metric, along with the demonstration of distinguishing between spoofing and multipath events, is presented for a set of test cases.
  - **Chapter 5** introduces the observation of the Automatic Gain Control (AGC) gain as a useful method for spoofing detection. The Chapter is based on the use of outputs of commercial receivers in order to detect spoofing attacks. These outputs, which include the AGC gain and a set of correlator values, are used to detect anomalies in the correlation function by means of an SQM test and to observe the total power on the receiver's band by means of the AGC gain. Thresholds for each metric are calculated by means of real datasets collected using the receiver, and they are then tested against meaningful scenarios. Finally, the concern of false alarms in the AGC measurements, given the presence of other types of radio-frequency interference (RFI) is addressed by a cross-observation of the AGC gain and the Carrier to Noise density ratio (C/N_0).
- **Part III: Spoofing Mitigation**

- **Chapter 6** introduces a novel mitigation technique, named the Time Jumper algorithm. This technique aims at detecting distortions in the correlation function by means of a linear decomposition of the signals. Thanks to this decomposition, the technique is able to estimate the delay difference between the spoofing signal and satellite signal, and with this information is able to unlock the receiver that is tracking the spoofer signal, and lock on to the satellite signal. In this way the algorithm is able to mitigate the effects of the spoofing attack. The technique is also aided by the presence of a Kalman filter that uses the unspoofed Doppler measurements in order to compute the correct position and not the spoofed one, obtaining an unspoofed position throughout the test.
- **Chapter 7** presents a spoofing detection algorithm using the outputs of a feedback tracking architecture, known as Extended Coupled Delay Lock Loop (ECADLL). This architecture was previously introduced for multipath mitigation and in this Chapter the feasibility for spoofing detection is shown. Along with the detection algorithm, validation of the proposed method is also presented, both for the estimation of the correct spoofing parameters and the correct distinction of spoofing events. The technique is compared in terms of detection capabilities with the metric introduced in Chapter 4, and numerical results of the implementation of the algorithm are presented. Finally, the mitigation of spoofing effects is demonstrated.

Summary and Conclusions and future work are drawn afterwards.

Appendix A provides a description of the different datasets used throughout this thesis, including the Texas Anti Spoofing Test Battery (TEXBAT), Wide Area Augmentation System (WAAS) data and dynamic data collection in urban environment from Turin, Italy.

Appendix B provides the mathematical derivation of the threshold computation for the Log-Likelihood ratio test.

Main Contributions

The main contributions of this thesis can be summarized as:

- Analysis and validation of a spoofing detection algorithm, based on ratio metric tests, using the Neyman-Pearson criterion and performance improvement by using an observation window.
- Description of a multidimensional ratio metric test for discrimination between spoofing attacks and multipath events. The new metric takes into account the ratio test, the temporal effects and the number of satellites, for this purpose.
- A novel cross-check between AGC gain and the C/N_0 for spoofing detection and the reduction of false alarms due to the presence of RFI events.
- A novel signal processing technique for spoofing mitigation purposes is introduced as the Time Jumper algorithm. Analysis of the different components and the validation of the spoofing mitigation capabilities are also shown.
- A spoofing detection algorithm, using the outputs of the ECADLL architecture is designed and validated against spoofing datasets. An example of the mitigation abilities of the ECADLL is presented.

The work of this thesis has been presented in different international conferences of worldwide renown in the GNSS community [5, 54, 55, 10] and in one published journal publication [53]. Three additional journal publications have been prepared and submitted and are currently under review process.

With the developments included in this dissertation, the GNSS receiver design may acknowledge the spoofing attacks as a critical point for its correct functioning, and by using the knowledge and techniques enclosed in this document, it might include different levels of protection for the new generation of GNSS receivers. In this way, the users of GNSSs may be warned and protected from spoofing attacks and may adapt the usage of the obtained information accordingly to its particular application.

Part I

Basics of Global Navigation Satellite Systems and the Spoofing Attacks

Chapter 1

Introduction to Global Navigation Satellite Systems

As the years pass by, the Global Navigation Satellite Systems (GNSS) are becoming an invisible technology, used by a big portion of the society, but one that is not fully understood by the typical user. As a consequence, the innovative uses and the possible threats to GNSSs are also unknown. The United States (U.S.) Global Positioning System (GPS) has been around for more than 20 years now, and people have adopted the use of navigation systems in everyday life, to the point where paper maps are becoming obsolete and everyone owns a GNSS receiver in some form. Knowledge of the basic operation of GNSS, and understanding of its limitations and risk, should be an important topic for the general user.

The goal of this Chapter is to present a condensed and brief summary on the GNSS functional basics and to introduce the knowledge needed to follow the discussions presented throughout this thesis. This Chapter is based mainly on the knowledge obtained during the PhD studies and on the analysis done in [45, 59, 13].

1.1 Fundamentals of GNSS

Ever since mankind started exploration of the world, localization has always played a critical role in the successes obtained. Initially, navigation was performed with drawn maps, using reference points like mountains, coves or rivers. As soon as oceanic

water navigation started, the lack of reference points had a big impact. Navigators started to rely on celestial observations to locate themselves. With the appearance of the science of timing and timekeeping, the accuracy level of celestial reading increased and up until today, timekeeping plays one of the most important roles in navigation systems.

To navigate could be defined as *to drive in a safe and secure mode a mobile from its starting point to its final destination*. With this concept in mind we can understand that navigation is a real time process and that it is important to retrieve the position of the mobile being moved and to keep measures consistent with the speed of the vehicle. Modern navigation no longer estimates its own position by observing the stars, but, in a similar fashion, is done by collecting radio signals transmitted by satellites at known positions. Position is then retrieved by doing trilateration of the different distances obtained, by processing the received signals.

1.1.1 Position, velocity and time

The final goal of any GNSS system is to allow the user to calculate its Position, Velocity and Time (PVT) solution. In order to obtain this solution we must first define some reference system for the position and timing. GNSSs use the Time of Arrival (TOA) concept to retrieve the PVT solution. This concept measures the time interval between the transmission of the signal from the satellite and the time when the signal is collected by the receiver. By means of this propagation time, the receiver is able to calculate the distances between the receiver and the satellites. For our three dimensional system we create a sphere that surrounds every satellite with radius equal to the distance obtained and observing the intersection of three of these spheres we can obtain our position. Assuming that the satellite sends the signal at time t_0 and we received at time $t_0 + \tau$, the radius of the sphere can be expressed as in (1.1), where c is the speed of light.

$$R_i = c\tau \tag{1.1}$$

In Fig. 1.1 we can observe an example of trilateration with three different satellites. If we assume that the receiver's clock is perfectly synchronized with the GPS time, the TOA calculation becomes trivial. Unfortunately, this is not the case for GNSS receivers. Then, signals obtained from the satellite will have a bias, due to the

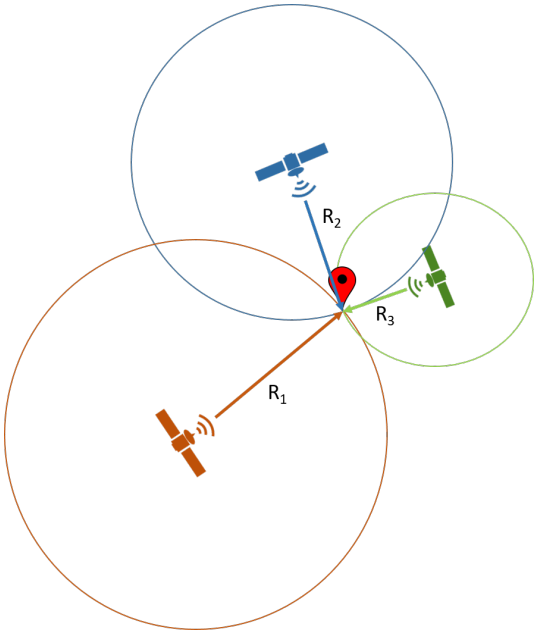

Fig. 1.1 Satellite trilateration example

difference between the GPS time and the receiver's clock time. The measurements performed by the receiver are called pseudoranges.

A general pseudorange is computed using (1.2), where x_u, y_u, z_u are the user's coordinates, x_j, y_j, z_j are the known coordinates of satellite j and $b = c\delta t_u$. A detailed description of the pseudorange computation is done in Section 1.4.4.

$$\rho_j = \sqrt{(x_j - x_u)^2 + (y_j - y_u)^2 + (z_j - z_u)^2} + b \tag{1.2}$$

1.1.2 Coordinate system

For every GNSS, it is important to define a position reference system and a time reference system. For global applications, since the earth surface is irregular and ever changing, the most common coordinate system is the ellipsoidal coordinates, where the earth is approximated as an ellipsoid of revolution, so for position P, we can define:

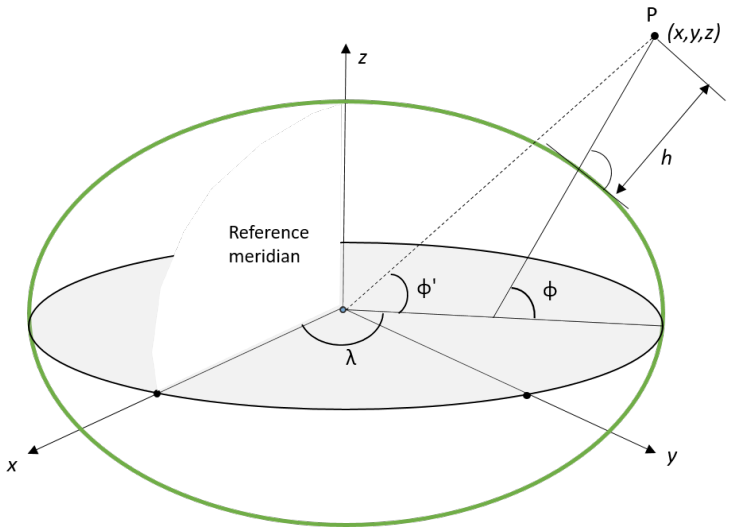

Fig. 1.2 Ellipsoidal coordinates (ϕ, λ, h) and Cartesian coordinates (x, y, z)

Geodetic Latitude ϕ is the angle in the meridian plane through the point P, between the equatorial plane and the line perpendicular to the surface of the ellipsoid passing through P.

Geodetic Longitude λ is the angle in the equatorial plane between the reference meridian and the meridian plane through P.

Geodetic height h is measured along the normal of the ellipsoid trough P

In Fig. 1.2 we observe a graphical representation of the ellipsoidal coordinates, compared to Cartesian coordinates (x, y, z). Cartesian coordinates are also commonly used in GNSS systems.

After defining the position coordinate system, we need to define the time reference system. In GNSS, where the position is obtained through time measures, it is crucial to have a precise synchronization between the clocks in each satellite. As observed in (1.1), if we have to multiply the time difference by the speed of light, a difference of one microsecond in satellite clock would translate to 300 meters of error in the range calculation. This means that the synchronization level in GNSS systems needs to be at the nanosecond level, if we require meter level precision.

1.2 Status of Global Navigation Satellite Systems

In this Section we introduce the major GNSS systems available at the time of writing and their status. We also introduce main differences between them that could interest the reader.

1.2.1 Global Positioning System

The U.S. GPS is a satellite based radio navigation system, developed and deployed by the U.S. government, that provides a global positioning service. In the 60's, when several U.S. government organization were interested in a navigation system that had global coverage, continuous operability and high accuracy, the investigations and construction of the different satellites began, and by 1973 the GPS project, basic architecture and number of satellites were approved. The GPS has been operational since 1978, even if its said that the civilian use of GPS was born when Selective Availability was turned off, in the year 2000 [59].

Today the GPS provides two services: the Standard Positioning Service for civil usage, and the Precise Positioning Service, for military and authorized government usage. Hereafter the signals provided by the standard positioning service, that are open and not encrypted, will be referred as civil signals, while the encrypted ones provided by the precise positioning service will be referred as military signals.

GPS Architecture

The GPS basic architecture is divided in three segments, the Space Segment, the Control Segment and the User Segment, as can be seen in Fig. 1.3.

Space Segment. The space segment is represented by the constellation of satellites from which the users are able to make measures. At the moment of writing, the GPS constellation consists on 31 in-orbit satellites. All the satellites have an altitude of 20,162.61 km from the surface of the earth and are synchronized using a rubidium or cesium atomic clock.

Control Segment. The control segment of the GPS consists of all terrestrial reference stations that coordinate activities between satellites, monitor the orbits,

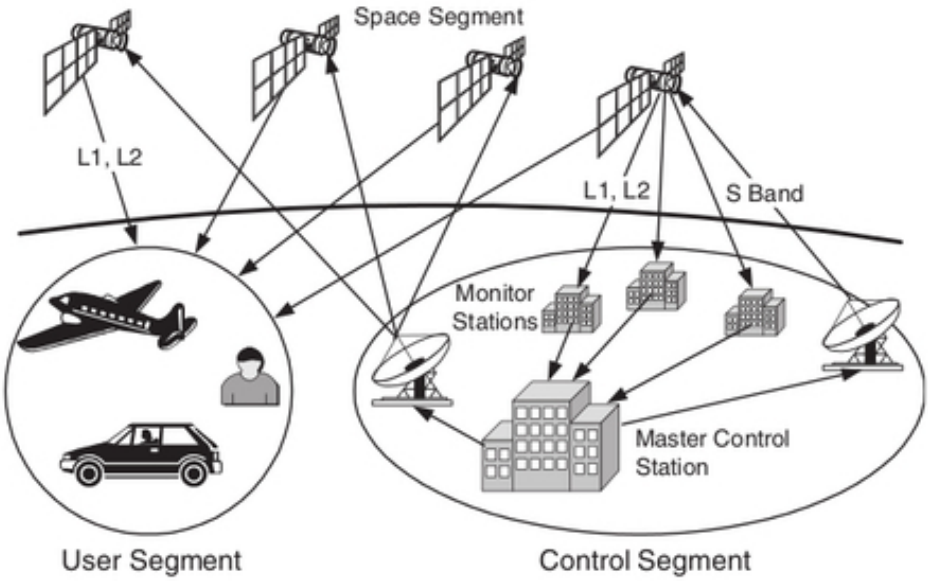

Fig. 1.3 GPS segments of the basic architecture

synchronize the atomic clocks and exchange information for the construction of the navigation message (ephemeris). It is composed by a master control station, located in Colorado, U.S., 16 monitoring stations located around the world, with four of them co-locating also the ground antennas for the communications with the satellites.

User Segment. The user segment is composed of all receivers that benefit from the GPS services. This segment has a wide range of realizations and applications. The evolution of receiver performance and size has been one of the key factors in the increasing usage of the GPS. In less than 20 years, the size has reduced from bulky, box-size receivers to micro-receivers that fit inside a cellphone. The accuracy has also improved, to under meter levels, in most occasions.

GPS modernization

The GPS has been undergoing modernization since the year 1990 in all three segments. The goal of modernization for the civil service is to improve accuracy, availability, coverage, integrity and robustness. In order to do so, modernization has introduced new civil signals (L2C, L5 and L1C) and frequencies (L5) in order to pro-

vide the user with increased redundancy, possibilities for ionospheric corrections and higher accuracy. The new signals have been sequentially introduced with the launch of new generations of satellites. A GNSS satellite lifespan is limited to 10-15 years, so they need to be substituted periodically. The newest satellites for GPS belong to the block III and will introduce a second civil signal in the L1 frequency (L1C) in the beginning of 2017. As of December 2016, the GPS constellation consists of 12 satellites available from block IIR, transmitting only the legacy signals, 7 from block IIR-M, transmitting also L2C signals, and 12 from block IIF, transmitting both L2C and L5 signals.

Other segments have been modernized, e.g. by the introduction of new signal processing techniques for receivers, improving the position accuracy. The improvements on the installations of the control segment, that introduced a backup control center and additional monitoring stations, improve the overall robustness of the system.

1.2.2 European GNSS (Galileo)

The European efforts on satellite navigation systems started with the Geo-stationary Navigation Overlay System (EGNOS), with the goal of improving actual GPS performances in Europe and to constitute the basis for future European satellite navigation development.

The European GNSS, named Galileo, was declared operational the 15th of December of 2016. Galileo is the first GNSS to be fully controlled by civil agencies and developed with civil applications in mind. It improves performances with respect to GPS legacy signals. Galileo is compatible with GPS but independent from it and also have inter-operability with GLONASS and other systems.

The basic configuration of Galileo envision 30 satellites, positioned in three different orbits, with a distance from the center of mass of the earth of 29601.897 km. It provides different civil services, depending on the typology of the final user. Three different GNSS services, plus one to support Search and Rescue, are envisioned. These services are:

- The Open Service (OS). Provides position and timing information, free of user charge, that competes in performance with other GNSS systems. The open service navigation data is broadcast in frequencies E1 and E5.

- The Public Regulated Service (PRS). Provides navigation and timing for authorized users, with high continuity and accuracy. PRS navigation data will be broadcast in frequencies E1 and E6.
- The Commercial Service (CS). Will give access to two additional signals, to improve accuracy and higher data throughput with a user fee. As December 2016, this service is not yet available.
- The Search and Rescue Service (SAR). Broadcasts globally alerts and distress signals received by the satellite and communicates back an acknowledgement signal.

At the time of writing, there are 18 available satellites in the Galileo constellation, 3 of which are unavailable at the moment due to technical difficulties. One of the major improvements of the Galileo signal over GPS, is the introduction of a new modulation called Binary Offset Carrier (BOC), which allows for better ranging precision and optimization of the frequency band allocation. Another improvement comes from the introduction of Hydrogen maser atomic clocks, that are much more accurate than rubidium and cesium clocks.

1.2.3 GLONASS system

The Russian GNSS, called GLONASS, has been operational since 1993 and achieved optimal status in 1995 with 24 satellites. When the Selective Availability was still applied to GPS signals, GLONASS looked very attractive to civil users in mid-90s. Positioning accuracy was better than GPS during these years, but it then collapsed as the system was not maintained properly. Lately, since year 2000, the Russian government has been working for the restoration of the system, updating their satellites and designing modern signals to be broadcast. At the time of writing, GLONASS has 24 operational satellites, 2 spares and 1 in flight tests. Since 2011, the latest generation of satellites, called GLONASS-K/KM has been deployed.

One of the main differences between GPS and GLONASS signals is that the latter use a Frequency Division Multiple Access (FDMA), in order to distinguish the signal coming from each satellite, instead of the Code Division Multiple Access (CDMA) as is used by GPS and Galileo. This means that each satellite transmits a

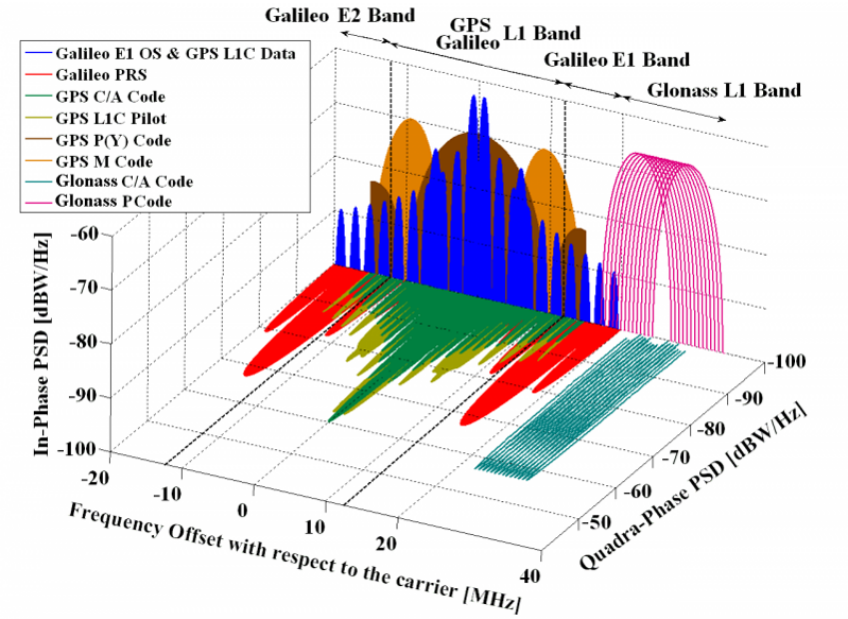

Fig. 1.4 Comparison of the frequency plot for L1, E1 and G1 band allocation, with the corresponding signals for GPS, Galileo and GLONASS. The impact of the FDMA is clear when assessing the GLONASS signal. *Obtained from GLONASS signals at www.navipedia.net*

signal in a slightly different frequency than it neighbors. The signals are separated by 500 kHz, and all satellites use the same maximal length code of 511 chips.

GLONASS is also undergoing an interesting modernization process, that will introduce CDMA signals into the system in order to improve accuracy. GLONASS signals are currently transmitted in two frequency, G1 and G2, and a civil signal and a military signal are transmitted in each one. In Fig. 1.4 we can observe the comparison between GPS, GLONASS and Galileo signal in the frequency band L1/E1 (GPS/Galileo) and G1 (GLONASS). In the figure, we can observe the different center frequencies of the FDMA modulation for the GLONASS signal (in purple and cyan lines) and the characteristic split spectrum of the BOC modulation for the Galileo signals (in blue and red).

1.2.4 Beidou system

Beidou is the Chinese GNSS and it was formerly known as COMPASS. Since year 2000 it has been providing limited navigation services, mainly in China, during the experimental setup phase.

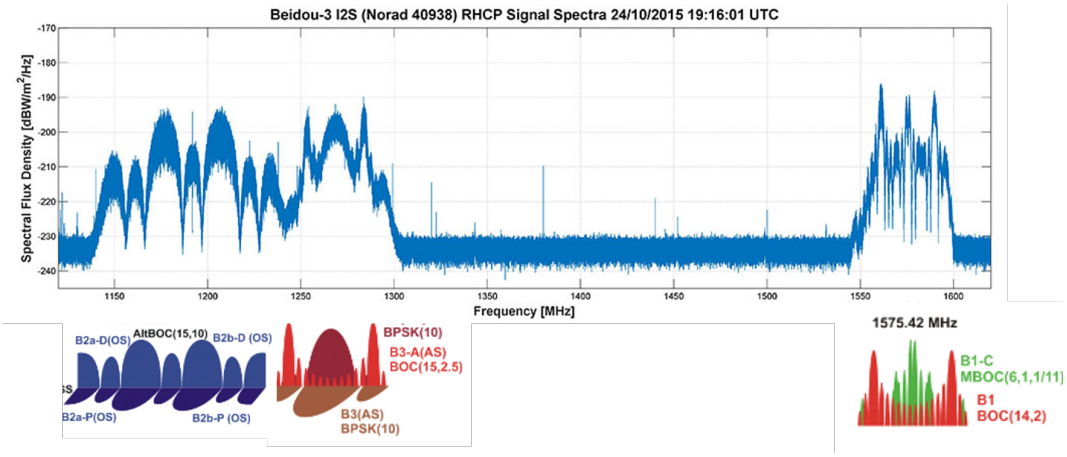

Fig. 1.5 Frequency plot of the signal coming from one of the IGSO Beidou satellites, along with a comparison against the theoretical shape of the signals. *Modified from [16]*

Since 2012, Beidou has provided regional navigation services for Asia-Pacific region. Currently 20 satellites have been deployed out of the 35 expected for the full constellation and among these, 6 of the satellites are allocated in Geostationary Orbits (GEO), 6 are allocated in Inclined Geo-Synchronous Orbits (IGSO) and the rest are allocated at Medium Earth Orbits (MEO). Beidou is the first GNSS to introduce the use of three different types of orbits among the satellites used for navigation. Starting from 2012, officially the construction of the last stage of the system began with the BDS-3.

According to the Beidou ICD [63], at the time of writing, satellites transmit two civil signals, one in B1 (1561.098 MHz) and another very similar in B2 (1207.14 MHz). On band B3, a QPSK(10) signal is transmitted, and is of reserved use. In Fig. 1.5, an interesting frequency plot, obtained by pointing an antenna at a Beidou satellite, from [16], is shown. In this plot, we can clearly observe the three signals broadcast by the satellite, along with a comparison with the theoretical shape of the signals.

1.3 GNSS signals

In this Section we present the basic structure of the GNSS signals and describe briefly their different components and characteristics. The signal structure is the core of the GNSS functionality and several characteristics will be exploited through this thesis

by the spoofing detection algorithms. Normally the signal broadcast by the satellites is denoted as Signal In Space (SIS), and in GPS and Galileo systems, it is transmitted using CDMA. This allows the receiver to identify each satellite without ambiguity, by distinguishing each individual code. Every satellite transmits a combination of signals, based on the frequency allocation and on the provided services.

For example, the signal transmitted by GPS satellites on frequency L1 is:

$$s_{L1}(t) = \sqrt{2P_{C/A}}D(t)c_{C/A}(t)cos(2\pi f_{L1}t + \theta_{L1}) + \sqrt{2P_Y}D(t)c_{P(Y)}(t)sin(2\pi f_{L1}t + \theta_{L1}) \tag{1.3}$$

where the first part is commonly known as the civil signal and will be the focus of attention of this thesis. Decomposing the civil signal, it has:

- an amplitude ($\sqrt{2P_{C/A}}$) that is maintained constant among satellites
- the Coarse/Acquisition (C/A) code ($c_{C/A}(t)$) that is unique for each specific satellite, identified by its Pseudo Random Noise (PRN) number
- the carrier signal ($cos(2\pi f_{L1}t + \theta_{L1})$), with carrier frequency f_{L1} and phase θ_{L1}
- and the navigation data bits ($D(t)$)

The military signal is very similar, just the code ($c_{P(Y)}$) is different. The military code is encrypted in order to avoid spoofing attacks on the military signal. In Fig. 1.6 we can observe a decomposition of the signal. It includes the carrier signal (top), the code (middle) and the navigation data bits (bottom). The figure is very useful to understand the rates of each specific part, e.g. in GPS L1 civil signal, the carrier has a frequency of 1575.42 MHz, the code rate is 1.023 Mcps and the navigation data rate is 50 bps.

1.3.1 Frequency allocation

In order to assure better performance of the different GNSS signals, a careful consideration of the allocation of the band of transmission has to be done. The different carrier frequencies of GPS and Galileo are reported in Table 1.1. In Fig. 1.7 the complete frequency allocation for all the systems and frequencies is shown. This is a useful figure to understand all the signals available from different GNSS and the

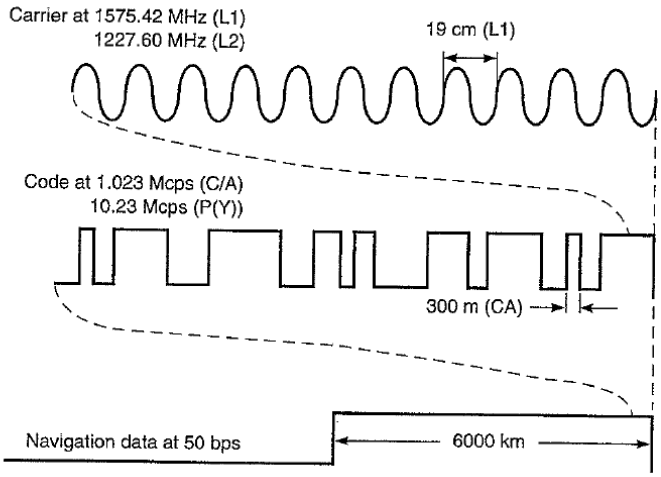

Fig. 1.6 GPS satellite signals, containing the carrier signal (top), the code signal (middle) and the navigation data (bottom). *From [59]*

System	Band	Bandwidth (MHz)	Center frequency (MHz)
GPS	L5	24 [1164-1188]	1176.45
	L2	20 [1217-1230]	1227.60
	L1	24 [1563-1587]	1575.42
Galileo	E5a	27 [1164-1191.795]	11176.45
	E5b	25 [1191.75-1217]	1207.14
	E6	40 [1260-1300]	1278.75
	E1	32 [1559-1591]	1575.42

Table 1.1 GPS and Galileo frequency bands

ones to come in the near future. With all these signals available, a bright future for GNSSs is guarantee.

Observing Fig. 1.7' several key characteristics of the GNSS systems can be retrieved. For example, the GPS Interface Specification [25] indicates 3 different frequencies and 10 different signals. From these 10 signals, 2 are in L5 frequency, transmitted in-phase (L5-I) and quadrature (L5-I), 3 are in L2 frequency, 1 being the civil signal (L2C) and 2 military signals (modulated with P(Y) and M codes). Finally in L1 frequency, 5 different signal will be present, 3 civil signals (C/A, L1C-I and L1C-Q) and 2 military signals (P(Y) and M codes).

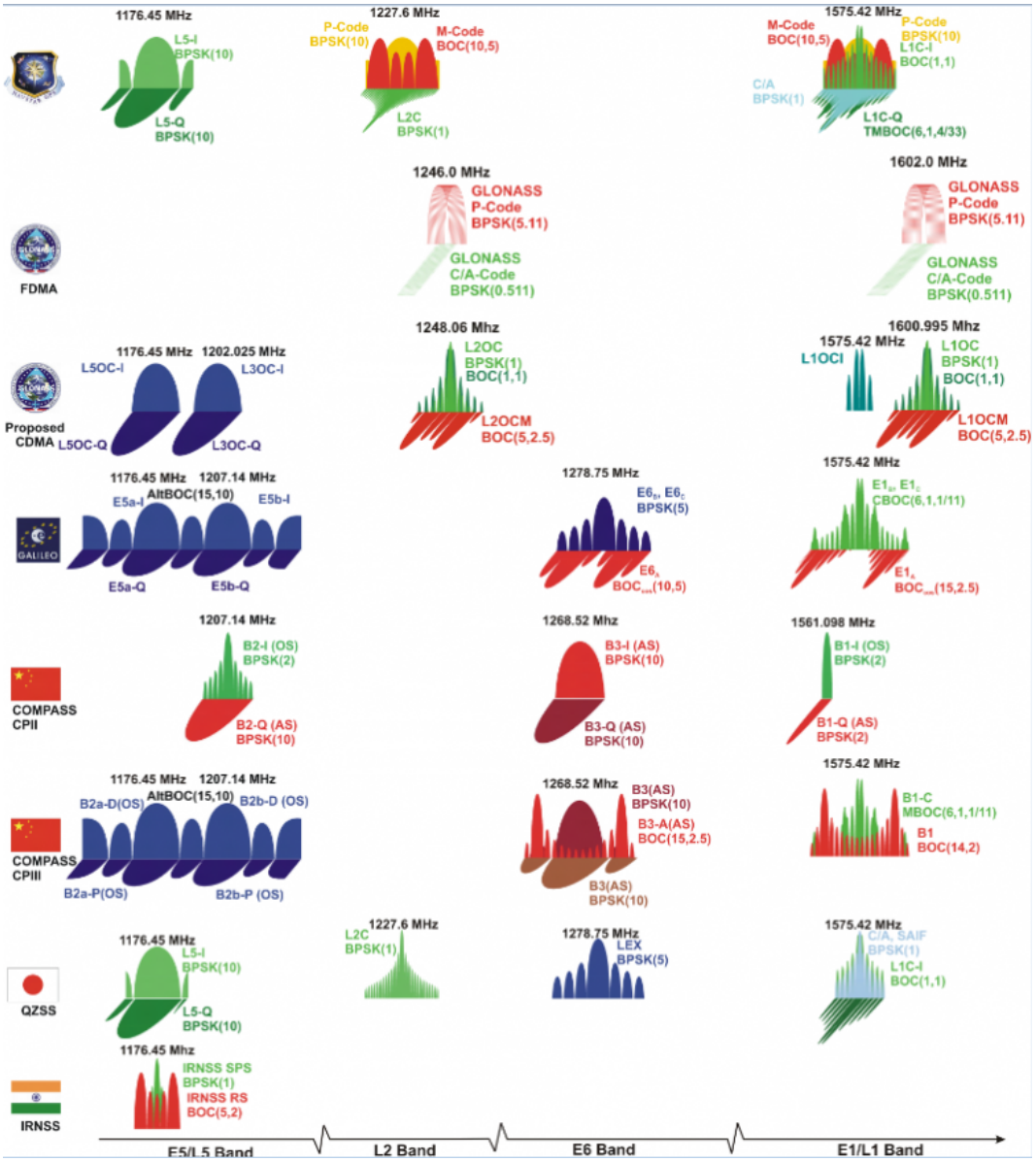

Fig. 1.7 Frequency allocation and spectrum of all the different GNSS signals, available and to come. From top to bottom, the systems are: GPS, GLONASS, Galileo, Beidou (China), QZSS (Japan) and IRNSS (India). *Obtained from GNSS signals at www.navipedia.net*

Not all of the signals present in Fig. 1.7 are transmitted at the time of writing, but they are on the plans of the modernization of the different GNSS. We can observe how the different codes and modulations change the spectrum appearance of each signal, in order to better utilize the available spectrum. It is important to notice that QZSS and IRNSS are regional systems that will not cover global navigation, but will only focus on their respective regions (Japan and India respectively).

1.3.2 GPS signals

In GPS, the satellite signal is a binary phase-shift keying (BPSK) modulation of a sequence of bits, with a rectangular shape. The code sequence can be generated by a maximal length linear feedback shift register. The need of efficient autocorrelation and low cross-correlation is imperative in order to distinguish between transmitted satellites. GPS uses a family of codes known as Gold codes that have these characteristics. Gold code sequences can be generated by using two different sequences of the same length. For generating GPS C/A code, the length of each maximal length sequence p=1023 (N=10) and the sequence for each satellite is chosen among the 1025 available Gold sequences. Each GPS satellite signal is characterized by a shifted version of the G2 polynomial (bottom), and the corresponding code is known as its unique PRN number. PRN numbers go from 1 to 32 in the GPS constellation.

In Fig. 1.8 we can observe the evolution of the GPS signals as they have been made available in the different blocks of satellite families. The first block III satellite, is scheduled for launch in the beginning of 2017.

As can be seen in Fig 1.8, two signals are transmitted by all available satellites, and they are known as legacy signals. The legacy signals include:

- C/A Code. The C/A code is used for the civil service and for the acquisition of the military devices. It is a bi-phase modulated signal with a chip rate of 1.023 Mcps, hence the chip duration is approximately $1\mu s$ and the code duration is of 1ms. It is a short duration code, that allows fast acquisition of the signal, even if not as precise as other codes. Only one signal uses C/A codes and that is the civil signal on L1. This C/A L1 signal is, by far, the most used signal as it is obtained and tracked by all GNSS receivers. It will be the main signal explored through this thesis.

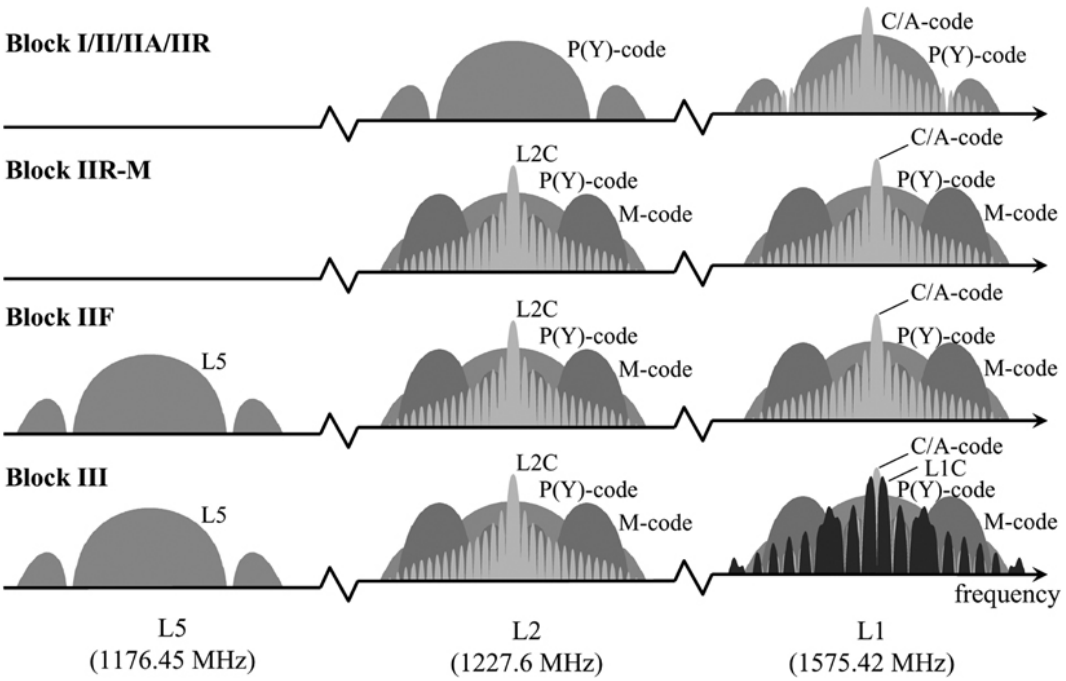

Fig. 1.8 Different GPS signal for the different Blocks. *From [59]*

- P(Y) Code. The protected code is a bi-phase modulated code at 10.23 Mchips/s. The shorter chip permits greater precision than with the C/A codes. The code length is of one week, and changes every week between the different satellites. It is a secure and encrypted code, and only authorized users have access to it.

Instead of repeating the C/A code, for the L2C civil signal, the signal contains two PRN sequences, the moderate length code and the long code. They are used alternatively, the moderate for the initial acquisition and the long for longer iterations and better performance. The L2C signal is currently only available in satellites from block IIR-M and IIF.

The message structure on the GPS is formatted into 30 bits words, and each word is grouped into sub-frames of 10 words that are 300 bits in length and 6 seconds in duration. Frames consist of 5 sub-frames and a completed message is a super-frame that consists of 25 frames, and is transmitted over 12.5 minutes. A full description of the GPS message structure can be found in [25].

The undergoing modernization of the GPS, as shown in Fig. 1.8, includes: the complete implementation of the L2C signal, the implementation of one more civil signal on L5, plus the transmission of Memory M-code signals in L1 and L2 that

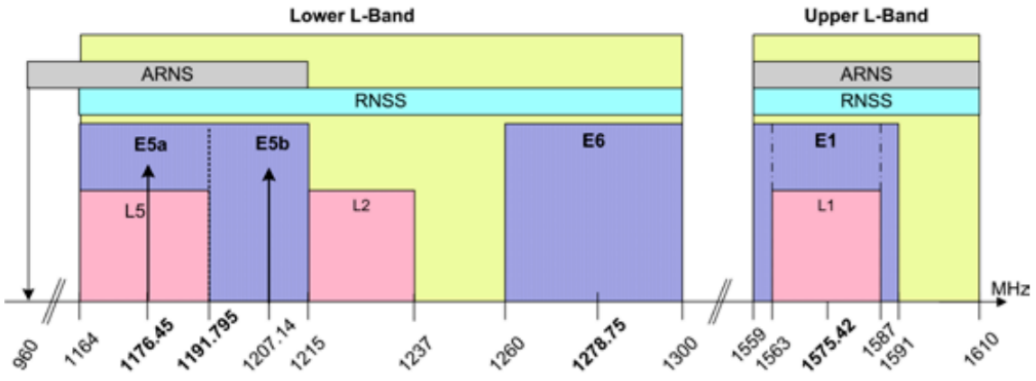

Fig. 1.9 Galileo frequency bands used (E) and the comparison with the GPS bands (L). *From [75]*

will modernize the P(Y) code signals, by using BOC modulation. For the GPS III block, the expectation is to have the fourth civil signal L1C, that is designed to maximize inter-operability with Galileo, and will use the Multiplexed BOC (MBOC) modulation [32, 6].

1.3.3 Galileo signals

For Galileo we will try to do a brief review of the signals that are designed for the different services provided. A full decryption of the Galileo SIS can be found in the Galileo ICD [75]. One of the most important things taken into account during the design of the Galileo signals, was to assure the inter-operability with the existing GNSSs, in particular with GPS. At the same time Galileo has to meet other requirements like independence, high performance, different services and security. A way to reduce inter-system interference is to use different modulations, for example modulating the code sequence with a subcarrier signal. This method is known as the BOC modulation.

Galileo satellites broadcast signals in 3 different frequencies, known as E1, E6 and E5 as observed in Table 1.1. The distinction between different satellites is done by CDMA as in GPS. In Galileo, the introduction of pilot channels will provide better and faster tracking for the signals. The pilot channels will be data free code transmission, that will help the receiver recover the data channel in a more robust way. In Fig. 1.9, the different bands used by Galileo are summarized.

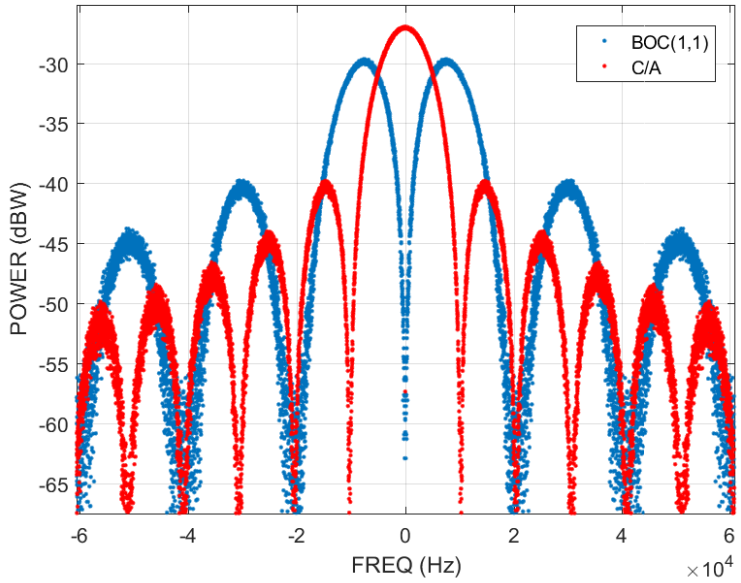

Fig. 1.10 Comparison of the frequency spectrum of the BOC signal (in blue) and the GPS C/A code signal (in red).

BOC modulation

The BOC modulation uses a subcarrier binary signal with a frequency n times 1.023 MHz and a chip rate that is m times 1.023 Mcps. With this notation, the modulation is written as BOC(n,m).

An interesting feature of the BOC modulation is that the maximum energy is not found at the center of the carrier frequency. The presence of sub-carrier modulation resolves in the characteristic split spectrum of the BOC modulation, and the maximum energy will be at the center frequency plus and minus n times 1.023 MHz. Thus a BOC(1,1) signal, like the one shown in Fig. 1.10, will have the maximum peaks at $+/-1.023$ MHz w.r.t. the carrier frequency. The BOC modulation allows for sharper correlation function and better ranging accuracy as will be detailed in Section 1.4.3.

In Fig. 1.11 the different frequency spectrum of each of the Galileo signals are shown. We observe that for band E1, shown in (a), we have a CBOC(6,1,1/11) for the OS, and a BOC_{cos}(15,2.5) for the PRS in quadrature. In band E6, shown in (b), we observe a BPSK(5) for the CS in-phase and a BOC_{cos}(10,5) for the PRS in

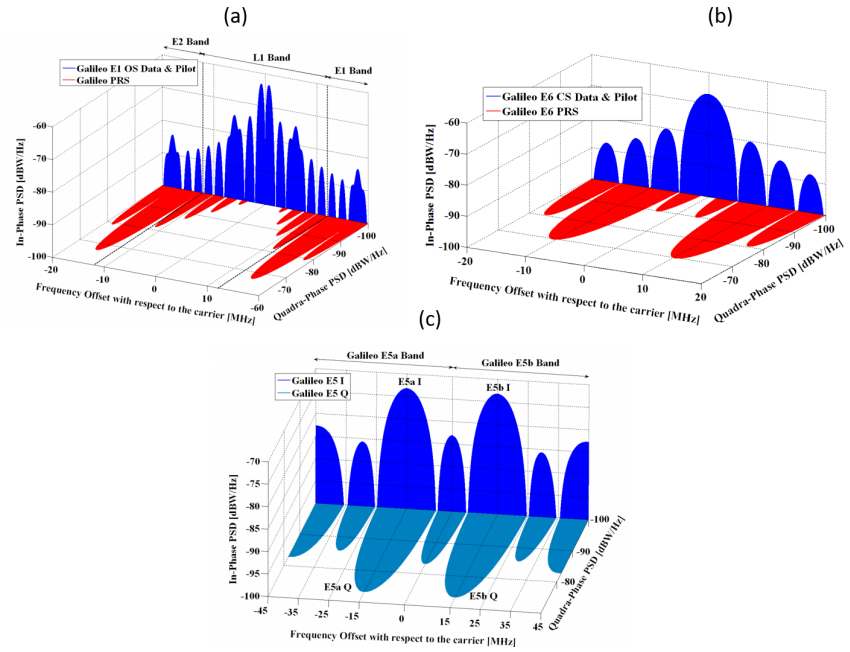

Fig. 1.11 Galileo signal spectrums in frequencies E1 (a), E6 (b) and E5 (c). *From Galileo Signals at www.navipedia.net*

quadrature. Finally for band E5, shown in (c), the AltBOC(15,20) is shown for I and Q channels.

1.4 GNSS receivers

GNSS receivers are used in order to collect the signal coming from the satellite, process it and compute a PVT solution. In this section a brief explanation of the functionality of a GNSS receiver is provided and the general GNSS receiver architecture is described. The typical receiver architecture has several blocks that work jointly to obtain the PVT solution, as shown in Fig. 1.12. The high-level blocks are the Radio Frequency (RF) Front End, the acquisition stage, the tracking stage and the PVT solution computation.

The basic functions of a GNSS receiver are: to capture and separate the SIS transmitted by the satellites, compute the pseudorange for each satellite by means of a TOA measurement, demodulate the navigation message to obtain ephemeris and

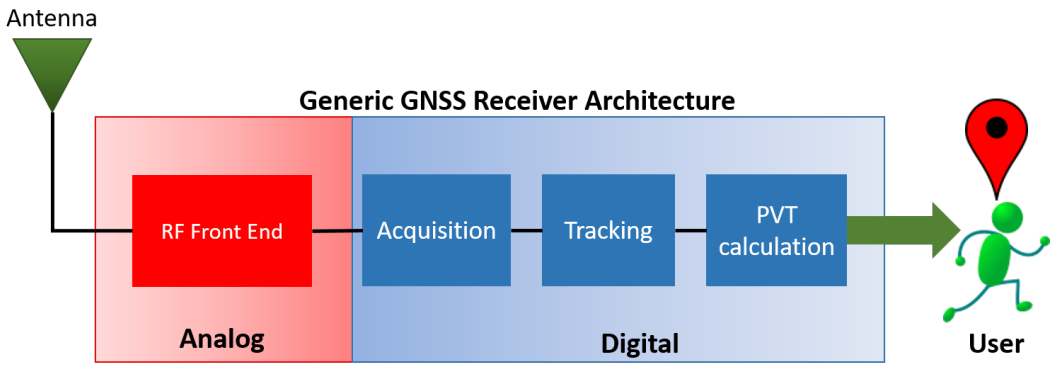

Fig. 1.12 Typical GNSS receiver architecture

estimate the PVT solution. Throughout this Section we will focus on the general architecture of a receiver able to track the GPS L1 civil signal.

1.4.1 Antenna and Front-End

The front-end includes all analog devices of the receiver and is in charge of receiving the satellite signal, performing a first level of processing, down-converting the RF signal to Intermediate Frequency (IF), filtering the signal to the desired band and converting it to digital in order to be processed by the acquisition stage. As shown in Fig. 1.13, the front-end is composed by the antenna, an RF filter and an amplifier stage, the IF down-conversion, the IF band-pass filter, the Automatic Gain Control (AGC) that optimizes the gain according to the Analog-to-digital Converter (ADC) dynamic range. Finally, the ADC converts the analog signals to its digital representation.

In general the complexity of the front-end presents a trade off that is proportional to the quality of the signal processing performed. Being the first stage of the process, the quality of the signal delivered by the front-end will improve or degrade all the receiver results. The input of the front-end is the RF analog signal transmitted by the satellite and the output will be the digitized version of the IF signal. This digitized signal is the input for the acquisition stage.

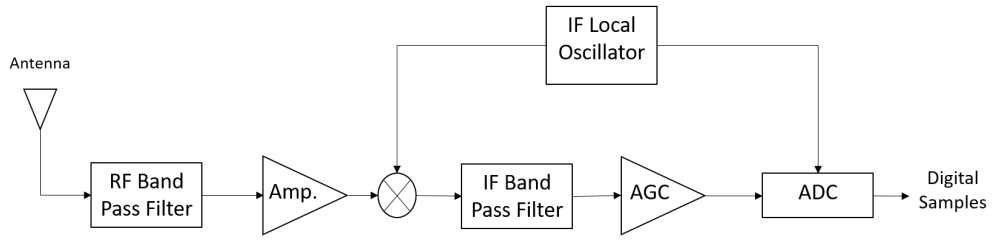

Fig. 1.13 Schematic of a generic GNSS receiver front-end. The input of the front-end is an analog signal and the output is the digital representation of such signal

1.4.2 Acquisition stage

The acquisition stage is a fundamental part of every GNSS receiver. The goal of the acquisition is to identify which satellites are in view, and obtain rough estimates of the code delay and Doppler shift of the received signal, in order to track and decode the information received.

A GNSS receiver, in order to track and decode a satellite signal, must be able to search over a set of PRN numbers, Doppler frequency ranges and code delays. The final goal of the acquisition stage is to estimate two parameters, for each satellite signal present, so the tracking loop can refine these estimates and obtain an accurate solution. The two parameters that need to be estimated by the acquisition are the delay of the code τ, and the Doppler frequency shift f_D. To be able to obtain a positioning estimate, the receiver must acquire at least four signals from as many different visible satellites. Each of these signals will have a different code delay τ and a different Doppler shift due to the relative distance and difference in speed between the receiver and the satellite.

After the front-end stage, the IF signal from the k satellite can be written as:

$$y(t) = a_k D(t-\tau_k) c_k(t-\tau_k) cos(2\pi(f_{IF} + f_{D_k})t + \theta_k) \tag{1.4}$$

where a_k is the amplitude of the signal after the ADC, $D(\cdot)$ is the navigation data bit, $c_k(\cdot)$ is the PRN code of the satellite, with delay τ_k, and f_{D_k} is the Doppler frequency introduced in the carrier signal.

The estimation of the the pair (τ, f_d) is based on the Cross Ambiguity Function (CAF), that is defined as the correlation between the incoming signal and the local replica of an individual PRN code. The CAF is a two-dimensional function in delay and Doppler domains that allows for the estimation of the parameters. In order to

obtain a high value in the correlation function of the codes, the carrier signal needs to be removed completely. If f_D is an unknown, this removal will not be possible. In this case a search over possible values for f_D needs to be done. Given the geometry of the satellites, the Doppler shift is usually in the range from ± 5 kHz from the L1 carrier frequency.

A GNSS receiver uses two reference signals during this process,

$$cos(2\pi(f_{IF} + \hat{f}_D)t + \hat{\theta}) \tag{1.5}$$

$$sin(2\pi(f_{IF} + \hat{f}_D)t + \hat{\theta}) \tag{1.6}$$

where they are used for the in-phase and quadrature processing respectively. If we multiply these signals by (1.4), we obtain

$$aD(t-\tau)c(t-\tau)cos(2\pi(\Delta f_D)t + \Delta\theta) \tag{1.7}$$

$$aD(t-\tau)c(t-\tau)sin(2\pi(\Delta f_D)t + \Delta\theta) \tag{1.8}$$

where

$$\Delta f_D = f_D - \hat{f}_D \tag{1.9}$$

$$\Delta\theta = \theta - \hat{\theta} \tag{1.10}$$

After this initial process, usually known as the carrier wipe-off we need to compute the correlation function of the code, by multiplying it by a local replica and integrating over an amount of time, where we assume that the navigation data bit is constant. This correlation function, for the in-phase branch, is built as:

$$R_I = \frac{aD}{T_{co}} \int_0^{T_{co}} c(t-\tau)c(t-\hat{\tau})cos(2\pi(\Delta f_D)t + \Delta\theta)dt \tag{1.11}$$

and an analogous equation can be found for the quadrature branch

$$R_Q = \frac{aD}{T_{co}} \int_0^{T_{co}} c(t-\tau)c(t-\hat{\tau})sin(2\pi(\Delta f_D)t + \Delta\theta)dt \tag{1.12}$$

Finally, we can construct the CAF function as the square sum of each correlation

$$R(\Delta\tau,\Delta f_D) = R_I^2 + R_Q^2 = \frac{a^2}{T_{co}} \int_0^{T_{co}} c(t-\tau)c(t-\hat{\tau})e^{j(2\pi(\Delta f_D)t)}dt \tag{1.13}$$

We can observe that the CAF function in (1.13) is only dependent on Δf_D and $\Delta\tau = \tau - \hat{\tau}$, that are the two parameters that need to be estimated by the acquisition stage. The integration time T_{co} needs to be lower than the navigation data bit rate, i.e. lower than 20 ms for GPS C/A code. This is due to the fact that a change on the data bit will hinder obtaining correlation results, because it would substract from the accumulated energy in the summation, instead of adding to it. The integration process can be performed over consecutive blocks in order to eliminate the possibility of having a data bit transition in every block.

In the search space of the CAF, the receiver needs to find the maximum value for both τ and f_D. There are several acquisition techniques reported in literature, that implement different types of searches and in general have a trade-off between the complexity of the search and the numbers of operations: two examples are the *serial search* and the *parallel search*.

For the GPS C/A signal, a serial search can be performed for all the possible values of the Doppler shift with a resolution of 500 Hz, and due to the short duration of the code, all the possible delays of the code are searched with a resolution of half a chip. We can observe how the number of computations increases rapidly with longer codes, as the one in the military signals, and it soon becomes unfeasible. On the other hand, the parallel search uses the convolution properties of the Fast Fourier Transform (FFT) of the signal to reduce the number of computations. In the software receivers used for this thesis, parallel search by means of an FFT is used [13]. In Fig. 1.14 a graphic example of a CAF is presented. There we can observe how the peak is at a clear combination of the Doppler shift and the code delay, meaning that a signal is present for said PRN number.

After the acquisition phase has estimated the values of $(\hat{\tau},\hat{f_D})$ it delivers these values to the tracking loop, that will be the one to refine the values in order to calculate the PVT solution and hence the user position.

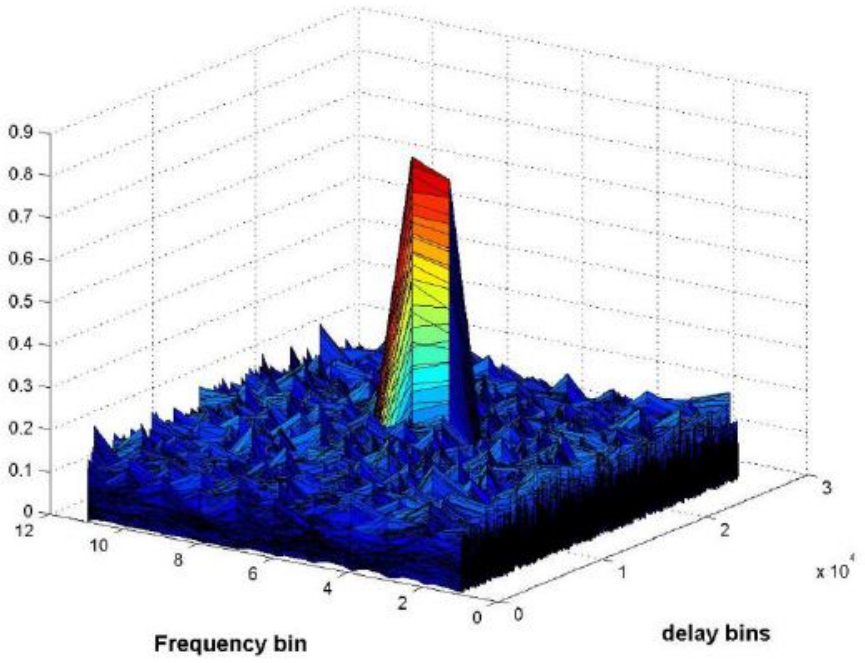

Fig. 1.14 Cross-Ambiguity Function example

1.4.3 Tracking stage

After the acquisition stage has roughly synchronized the receiver a fine synchronization takes over, continuously maintaining the lock and correcting the alignments by means of closed loop operations. As was stated in Section 1.4.2, the resolution of the acquisition results are usually of 500 Hz for the Doppler shift and of half a chip for the code delay. These resolutions need to be improved in order to obtain accurate positioning, and it is performed on the tracking stage. The tracking is performed over a double loop, in which it tracks the code delay and the carrier frequency of the signal. It uses a Delay Lock Loop (DLL) for the code and a Phase Lock Loop (PLL) for the carrier. The two loops are initialized by the outputs of the acquisition phase $(\hat{\tau},\hat{f}_d)$.

It is not possible to obtain a good estimation of the delay if all the residual Doppler shift is not removed, and it is not possible to obtain a good estimation of the residual Doppler shift until the code signal is not totally aligned. For these reasons the best estimates require several steps of approximation where the carrier loop is used to remove the Doppler modulation (carrier wipe-off) and the output of the code

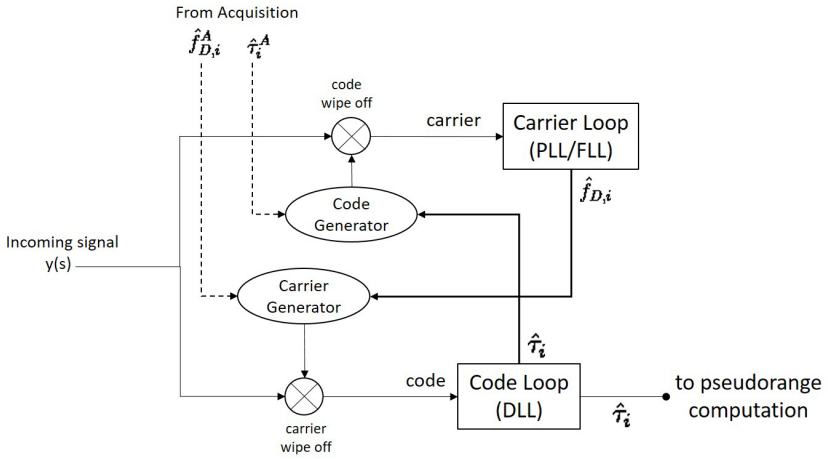

Fig. 1.15 General block scheme of the tracking loop, including the two branches of code and carrier information, and the initial information coming from the acquisition stage

loop is used to align and cancel the code signal (code wipe-off) [13]. In Fig. 1.15 we can observe a general architecture of the tracking loop.

The phase lock loop

The carrier tracking is performed in a feedback loop able to finely estimate the frequency of a sin wave, and to track its changes. For this loop a PLL is used. It is important to notice that after the carrier wipe-off has been performed, the signal is still modulated by the navigation data, so a PLL insensitive to phase transitions has to be adopted. In some cases, a Frequency Lock Loop (FLL) is used to track the initial frequency and after that, the PLL recovers the phase. Normally a Costas loop implementation is used for carrier tracking [45]. The goal of the Costas loop is to keep all of the energy in the I phase (in-phase) arm of the loop [13]. In Fig. 1.16 we can observe the general block diagram of a Costas loop PLL.

A similar analysis to what was done in Section 1.4.2 can be done for the PLL and the DLL of the tracking loop. After the code wipe-off and assuming an unitary amplitude, for the PLL we have:

$$y(t) = D(t-\tau)cos(2\pi(f_{IF}+f_D)t+\theta) \tag{1.14}$$

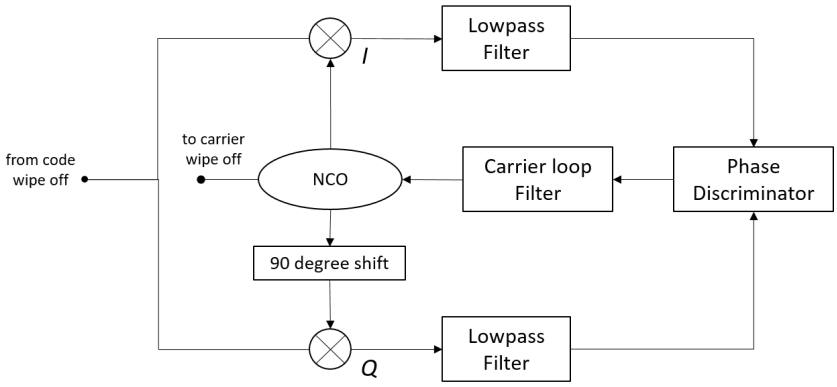

Fig. 1.16 General block scheme of a Costas carrier phase lock loop

if we multiply (1.14) by the locally generated carrier signals, we obtain:

$$y(t)cos(2\pi(f_{IF}+f_D)t) = \frac{1}{2}D(t-\tau)cos(\theta)+\frac{1}{2}cos(4\pi(f_{IF}+f_D)t+\theta) \quad (1.15)$$

$$y(t)sin(2\pi(f_{IF}+f_D)t) = \frac{1}{2}D(t-\tau)sin(\theta)+\frac{1}{2}sin(4\pi(f_{IF}+f_D)t+\theta) \quad (1.16)$$

for the in-phase and quadrature branch respectively. After the low pass filter, we obtain:

$$I = \frac{1}{2}D(t-\tau)cos(\theta) \quad (1.17)$$

$$Q = \frac{1}{2}D(t-\tau)sin(\theta) \quad (1.18)$$

Following, in order to recover the phase of the signal we can then use a discriminator such as:

$$\frac{Q}{I} = \frac{\frac{1}{2}D(t-\theta)sin(\theta)}{\frac{1}{2}D(t-\theta)cos(\theta)} = tan(\theta) \quad (1.19)$$

and with the results obtained from this discriminator it is possible to control the carrier generator. Other discriminators can be used for the Costas PLL, and detailed information of each of them and their performance analysis can be found in [13, 45].

After recovering the phase of the signal, the NCO carrier generator is able to generate an accurate replica of the incoming signal, which is needed for the recovery of the code delay by means of the DLL.

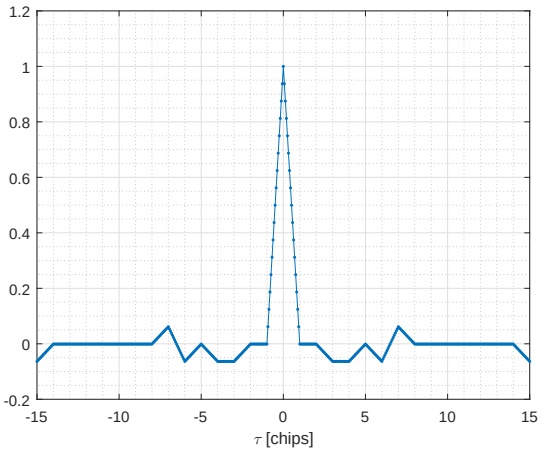

Fig. 1.17 Example of the correlation function for the GPS C/A code. The value of τ is the delay difference in chips between the local code and the incoming signal

The delay lock loop and the correlation function

The code tracking is performed by a DLL that is able to estimate the residual code delay. The information about the relative delay between the incoming signal and the local code replica is contained in the correlation function between the two. Therefore the goal of the DLL is to estimate the correlation function with high accuracy.

After the complete removal of the carrier signal and considering a unitary amplitude, we have a signal that looks like:

$$y(t) = D(t-\tau)c(t-\tau) \tag{1.20}$$

If we then multiply this signal by the local code replica, with delay $\hat{\tau}$, and we integrate over a certain time where we assume that the navigation data bit is not changing, we obtain the correlation function:

$$R(\Delta\tau) = \frac{D}{T_{co}} \int_0^{T_{co}} c(t-\tau)c(t-\hat{\tau})dt \tag{1.21}$$

This correlation function R is only dependent on $\Delta\tau$ and on the code structure. For the GPS C/A code, the correlation function assumes a triangular shape, as shown in Fig. 1.17.

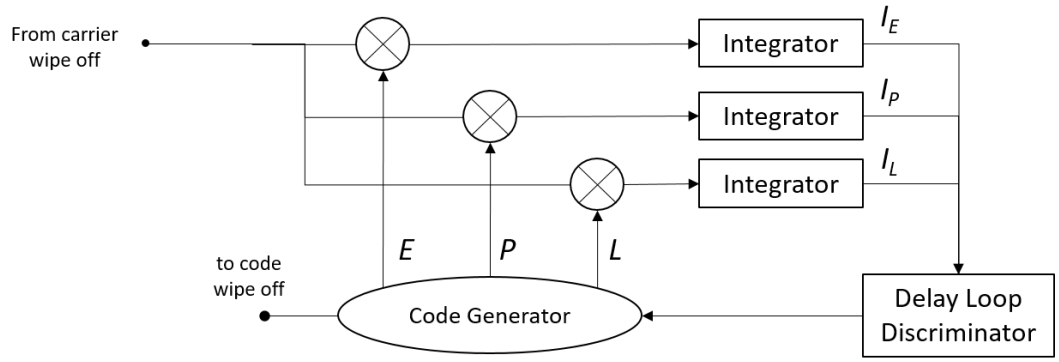

Fig. 1.18 General block scheme of the Delay lock loop, using only the In-phase branch of the signal

As can be seen from Fig. 1.17, correlation function R, is maximum when the two codes are completely aligned which is the desired condition for recovering the navigation data (D). However, a normal peak-search approach is not possible because it is dependent on the absolute peak value. In general the used approach is through a discriminator function that will be null only when the two codes (local and incoming) are aligned. In order to build this discriminator function, different code replicas, with different delays need to be generated. Generally, three different code replicas are built, and thus three correlator values are obtained, denoted usually as I_E, I_P and I_L, for the early, prompt and late replicas of the code. A general block scheme of the DLL is shown in Fig. 1.18, where the in-phase branch can be observed.

The DLL usually generates three different local code replicas, delayed between them by a fixed spacing, e.g. 0.5 chips. The DLL can then discern whether the two signals are aligned or not by means of a discriminator. This discrimination function is unambiguous with respect to the delay, contrary to the normal correlation function. It is proportional to the difference of the values of the early and late correlators and it is the input to the controller. An example of a possible discriminator function is the non coherent normalized dot product, constructed as:

$$\frac{I_D \cdot I_P + Q_D \cdot Q_P}{I_P^2 + Q_P^2} \tag{1.22}$$

where I_D and Q_D are the early minus late difference of each correlation branch. This discriminator function will be the one used in Chapter 7. Other examples of

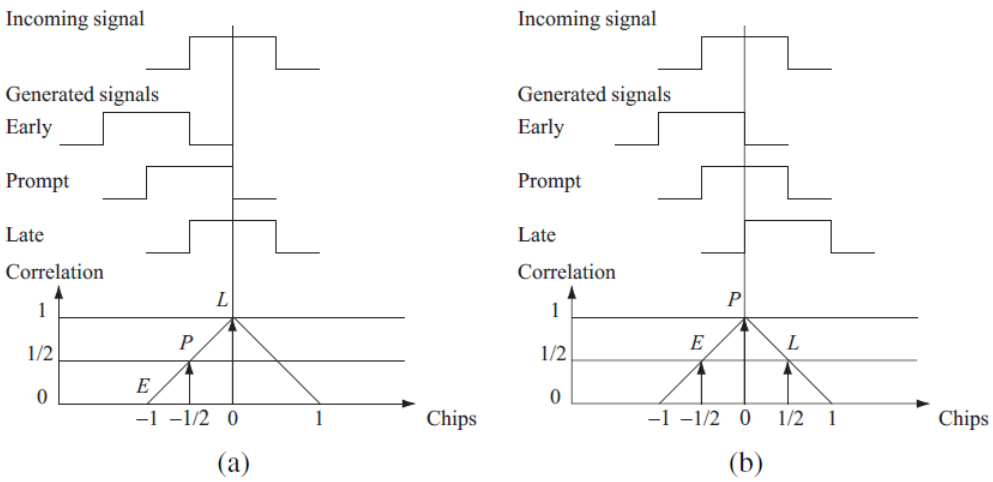

Fig. 1.19 Example of the correlation function results between two code replicas. In (A) the replicas are not aligned and we observe how the Late correlator is higher than the Prompt and the Early. In (B) the two codes are aligned and the Prompt correlator is at its maximum and the Early and Late correlators have the same value. *From [13]*

discriminator functions can be found in [45, 13], where a detailed explanation is presented.

The space where the signals are aligned in frequency, and only the delay difference between the codes is observed, will be referred hereafter as the correlation space. In Fig. 1.19 we can observe the process performed by the DLL, where three replicas are generated and based on the values of the Early, Late and Prompt correlators, the alignment of the signal is identified.

As stated before, the shape of the correlation function is dependent on the code structure and its modulation. As an example, one of the most important features of the BOC signal is the auto-correlation function resulting between two such sequences. A comparison of the correlation functions is shown in Fig. 1.20. The BOC autocorrelation function can potentially give better accuracy due to the sharper peak. But, due to the existence of the side peaks, it presents a trade-off between the higher accuracy, the complexity of the receiver and the possibility of false locks.

A good tracking stage is essential to every receiver, in order to obtain accurate pseudorange measurements. In fact the calculation of Δt can only be performed, if the receiver and the incoming signal are perfectly synchronized.

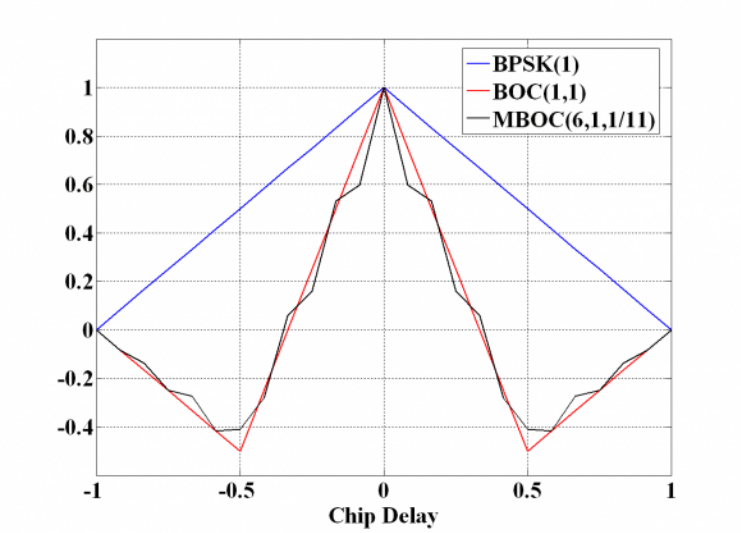

Fig. 1.20 Difference between BPSK and BOC autocorrelation functions.

1.4.4 Position Velocity and Time

The last step of a GNSS receiver process is to compute the pseudorange measures and obtain a PVT solution. The values of the pseudoranges are calculated using the delay observed by the tracking loop and information obtained by the decoding of the navigation message. Using these pseudoranges, a set of equations can be built in order to calculate the user's PVT solution.

Pseudorange computation

For the calculation of the pseudoranges it is important to recall that the GPS navigation message structure is organized in pages, and each page contains 5 subframes, and each subframe contains 10 words of 30 bits each. For the pseudorange calculations the first two words are essential, the Telemetry word (TLM) and the Hand Over word (HOW). The TLM is used to synchronize the navigation message and the HOW contains the information of the Time of Week (TOW), among other data. The structure of the TLM and HOW words is shown in Fig. 1.21.

In the navigation message, the satellite time is transmitted in the form of a Z-count that is specified at the beginning of each subframe. A single increment in the Z-count is equivalent to 1.5 seconds, and between two subframes the Z-count

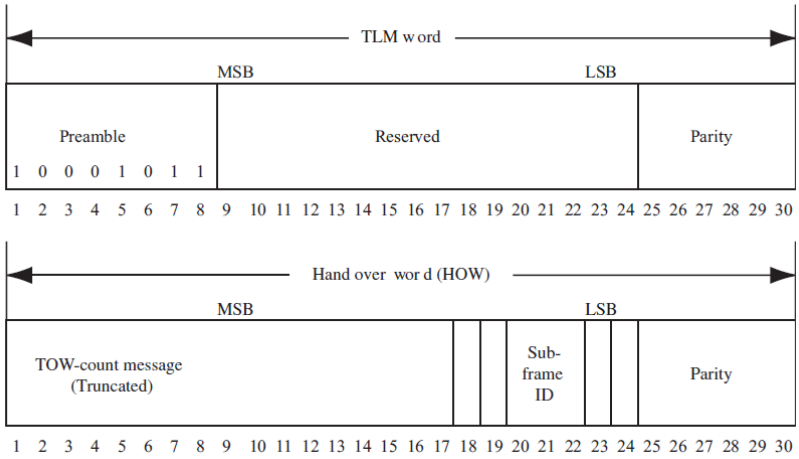

Fig. 1.21 The first two words of the navigation message, the TLM (top) and the HOW (bottom). *From [13]*

increments by 4 (i.e. 6 seconds). In order to determine the satellite clock time, we need the Z-count in the current subframe and the amount of time elapsed since the beginning of the subframe. The elapsed time is computed by counting the whole number of navigation data bits transmitted since the beginning of the subframe, plus the whole number of code periods since the beginning of the current navigation data bit, plus the number of chips elapsed in the current code cycle, plus the fraction of the current chip [59]. In Fig. 1.22, an example is shown that can clarify this concept. The number of navigation data bits and the number of code periods are computed by the PVT computation module, while the DLL provides the information required for the last two terms. By means of this time calculation and knowing the position of the satellites, we can obtain the pseudorange values ρ, by means of (1.2).

Finally, in order to solve for the unknown position and time (x, y, z and δt) of the receiver, a system of equations containing at least 4 pseudorange equations needs to be solved. The approximate satellite position is known due to the almanacs of the orbits, but the errors in these are usually too large to be used for obtaining satellite positions. Thus, more accurate positions are broadcast in the navigation message by each satellite. That means that reading the navigation message, not only provides the timing information, but also delivers information on the satellite positions, needed to find the user's PVT solution.

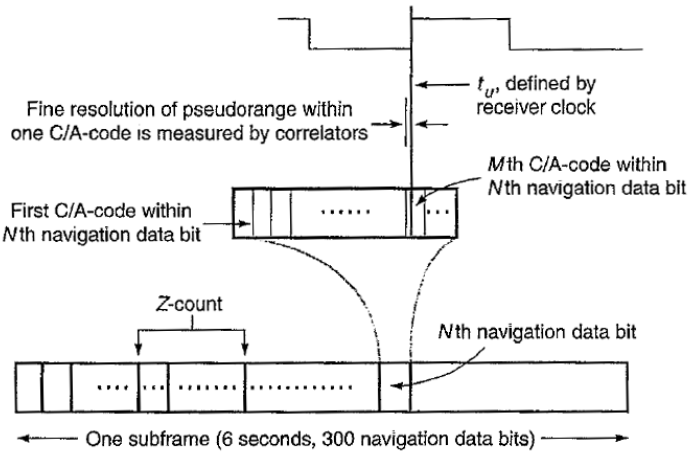

Fig. 1.22 Example of the pseudorange calculation using the different levels of data. The coarser resolution is that of the navigation data bit inside the subframe (bottom), then the code block inside that navigation bit (middle) and the finer resolution is the current chip inside the code block (top). *From [59]*

1.4.5 Position calculation by means of a Least Square solution

In order to obtain the user's position, we stated that we need to solve a system of equations containing the pseudoranges. To solve this system of equations, the most common technique is to use the *least squares* method.

This least squares method minimizes the mean square error of the solution of a system of equations, such as $A\mathbf{x} = \mathbf{b}$. The system is usually overdetermined, where A has m rows and n columns, and $m > n$. That means that there are more observations (m), than unknown parameters (n).

In the optimal least square solution, $\hat{\mathbf{x}}$ tries to minimize the error vector $\hat{\mathbf{e}} = \mathbf{b} - A\hat{\mathbf{x}}$. If we solve for the minimization of the sum of squared errors $||\hat{\mathbf{e}}||^2$, the solution is then obtained as:

$$\hat{\mathbf{x}} = (A^T A)^{-1} A^T \mathbf{b} \tag{1.23}$$

If we consider (1.2), and follow the procedure done in [13], we can obtain a linearized version of the pseudorange equation in vector form, as:

$$P^k = \rho_0 - \left[\frac{-(x^K - x)}{\rho_0^K} - \frac{-(y^K - y)}{\rho_0^K} - \frac{-(z^K - z)}{\rho_0^K}\right] \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \\ c\delta t \end{bmatrix} \tag{1.24}$$

for satellite K, and where ρ_0 is the pseudorange computed with the initial approximation of the user's position, and $\Delta X, \Delta Y$ and ΔZ are the updates of the user's position with respect to x_0, y_0 and z_0. The receiver's position at time zero is usually selected at the center of the earth (0,0,0) and a couple of iterations are required to obtain an accurate position at the current time.

Using (1.24), we can build matrix A as,

$$A\mathbf{x} = \begin{bmatrix} \frac{-(x^1 - x_0)}{\rho_0^1} & \frac{-(y^1 - y_0)}{\rho_0^1} & \frac{-(z^1 - z_0)}{\rho_0^1} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{-(x^m - x_0)}{\rho_0^m} & \frac{-(y^m - y_0)}{\rho_0^m} & \frac{-(z^m - z_0)}{\rho_0^m} & 1 \end{bmatrix} \begin{bmatrix} \Delta X_1 \\ \Delta Y_1 \\ \Delta Z_1 \\ c\delta t_1 \end{bmatrix} = \mathbf{b} \tag{1.25}$$

and solve for the position update $\Delta X, \Delta Y$ and ΔZ, using (1.23). Finally the position solution at time 1 will be obtained as:

$$\begin{aligned} x_1 &= x_0 + \Delta X_1 \\ y_1 &= y_0 + \Delta Y_1 \\ z_1 &= z_0 + \Delta Z_1 \end{aligned}$$

There are several errors that affect the pseudorange computation and that need to be taken into account for correct PVT estimations. These include the satellite's clock offset w.r.t. the system time, the tropospheric and ionospheric delays and the relativistic effects, among others. In general, the more precise the desired PVT solution, the more errors that need to be taken into account and corrected by the receiver. Detailed information on the PVT solution errors falls out of the scope of this Chapter, and can be found in [13, 45, 47].

The least square method is the most common way for obtaining PVT solutions in a GNSS receiver. Another commonly adopted solution is the use of a *Kalman*

filter for PVT computation. The Kalman filter takes into consideration the previous states of the system and the evolution of the solution using appropriate models and propagation methods in order to obtain accurate solutions. A general Kalman filter description is presented in Section 6.3, when it is used.

In this Chapter we presented a very general introduction to the satellite navigation systems, their basic architecture and the most commonly used signals. In Chapter 2 we introduce the spoofing attack concept, as well as an overview of the different anti-spoofing techniques that have been proposed.

Chapter 2

The Spoofing Threat

In a world where the technological innovations are advancing at an increasingly high speed, the related security aspects are elements of a growing concern. This is true also for those technologies based on the use of GNSSs, that, as well known, might be vulnerable to different types of RFI, both unintentional or of deliberate nature. In this Chapter we introduce the spoofing attack as a dangerous form of RFI, we describe the different types of spoofing attacks and present an overview of the different anti-spoofing techniques proposed in literature We finally discuss the differences between spoofing detection and spoofing mitigation concepts, making a case for the organization of this thesis.

2.1 Vulnerabilities of GNSS civil signals

Received GNSS signals have a very low amplitude once they hit the surface of the earth. The signal is usually well hidden below the noise floor of the receiver and signal processing has to be done by the receiver to recover the signal and use it for positioning purposes. This feature makes the signal vulnerable to any other transmission occurring in the GNSS bands. An additional signal present in the band, even if it has a small amount of power, could be enough for the receiver to lose track of the GNSS signal. GPS L1 and L5 bands are inside the Aeronautical Radio Navigation Service (ARNS) frequency bands, as observed in Fig. 1.9. This means that no other radio signal is transmitted in them and illegal transmissions are

heavily punished. Nevertheless, enforcement of these regulations is a difficult task as transmission in these bands can be easily accomplished.

In addition to the low power, the open nature of the signal makes it vulnerable to illegal transmission of counterfeit signals, that may be able to fool an unprotected receiver. With the increased availability of programmable simulators and software defined radio systems, the generation of counterfeit attacks is more feasible than ever [70]. These counterfeit generations are known as spoofing attacks and the design of techniques for protecting the receiver against them, is the main focus of this thesis.

2.2 Radio-frequency interference

Transmission of radio frequency signals, with the goal of disrupting the usage of GNSS systems, is becoming more common as the GNSS signals extend to new sectors of society and newer applications. The GNSSs broadcast their signals in frequency bands allocated in the L band as was introduced in Chapter 1. Transmission of unregistered signals in these bands is illegal and punishable in multiple countries.

Nevertheless, several GNSS applications are being used to control operations of companies and workers, becoming a desirable target for disruption signals. Different types of disruptions are possible, with different levels of effectiveness and sneakiness. As stated, the GNSS signal is usually received under the noise floor and the properties of the spreading code are used in order to recover the carrier. This characteristically low power makes it easy for an attacker to increase the noise level, thus making the GNSS carrier signal unrecoverable.

The simplest type of RFI is the *jammer* attack. This type of interference consists of the transmission of a high power signal in the frequency of the target GNSS receiver, with the goal of burying the satellite signal to levels were it cannot be tracked. This signal is not consistent with the satellite signal and its only goal is to disrupt usage of GNSS.

Another type of RFI is the so-called *meaconing* attack. It consist on the re-transmission of a recorded and delayed GNSS signal with a power higher than the satellite signal in order to make the receiver track the transmitted signal and obtain a position solution at the position of the attacker. These attacks usually result in a jump

of the user position and in inconsistencies between the time before and after attack, thus making it easy to detect if the appropriate checks are performed [23, 3, 45].

The final type of RFI is the *spoofing* attacks that will be described in Section 2.3.

2.3 Spoofing attacks

Spoofing attacks consist of the broadcast of a GNSS-like signal to sneakily take control of the receiver. The signal transmitted by the spoofing is aligned with the current constellation and if the spoofer is able to take control of the receiver, it can slowly modify the position solution without the receiver noticing any inconsistency or jumps in the solution. Throughout this thesis, the spoofing attacks are the main focus of the research, developing techniques for detecting spoofing and mitigating spoofing effects.

Many applications are desirable targets for spoofing attacks, especially if the goal is to perform illegal actions in secure positions or to sneakily disrupt correct functioning of the target receiver. For example, a fisherman may want to modify the position of a vessel, in order to access illegal waters without raising the warning of the geofence. Alternatively, an attacker may want to modify the position of a drone flying nearby, in order to make it crash and then be recovered by the attacker.

The effects that a spoofing attack can produce are well known [36, 23, 42, 61, 76, 30, 86], and its feasibility, also eased by the advances in the software defined radio (SDR) technologies, has been demonstrated and reported in recent literature [68, 2] and no longer solely in the domain of GNSS experts [34].

Nevertheless, a large number of applications based on the use of GNSS, can be considered critical in terms of *safety* and *liability* and, consequently, they have stringent requirements of *trustworthiness*. As examples, not only systems for transportation and fleet management rely on the position obtained from satellite navigation signals, but also power grids, telecommunications networks, financial transactions, etc. use GNSSs for synchronization purposes [23, 42].

Along this thesis, we focus our attention on external spoofing, meaning that a third-party user is trying to interfere with the GNSS receiver and it has not direct access to the antenna. This consideration does not take into account the cases where an individual with access to the receiver may disconnect the antenna cable or shield

it from the sky, in order for the receiver to track only the signals generated by him. These types of self-attack are not considered here, and can be detected by different methods, like discontinuity checks or bit synchronization checks, that are not explored in this thesis. The detection algorithms presented hereafter consider only the case in which a spoofing signal is combined with a true signal from the GNSS satellites.

In general, in order to take control of the Navigation solution a spoofing attack will try to modify the delay information of each satellite. In this way, the pseudorange calculation can be altered to obtain the position desired by the attacker. The way the delay information is used is dependent on the PVT algorithm inside the receiver, which may include more advanced algorithms like Kalman filter based solutions or other weighted combinations. This means, that unless the attacker has complete knowledge of the algorithms inside the receiver, the modification of the delay may not generate the desired final navigation solution. Taking into account these remarks, throughout the thesis we assume that the spoofer will always try to align the Doppler information of the spoofing and the satellite signals, so It can have a larger and more controlled impact on the DLL information.

2.4 Types of spoofing attack

In recent research, different types of spoofing attacks have been classified based on the complexity of the spoofer, and on the difficulty of detecting spoofing attacks from a receiver point of view [37, 42]. In Fig. 2.1 a graphical representation of the types of spoofing attacks is presented.

2.4.1 Simplistic spoofing attack

A simplistic spoofing attack is based on the use of a GNSS signal simulator to construct the fake signal and transmit it in order to fool the receiver, as observed on the left of Fig. 2.1. This type of attack is very easy to realize, but it is also expensive and easily detectable by means of trivial techniques, given that a large signal power is needed in order to neglect the satellite signal and that the fake signal is not synchronized with the constellation. Usually this type of attack results in jumps in the PVT computations, or they are performed by first jamming GNSS

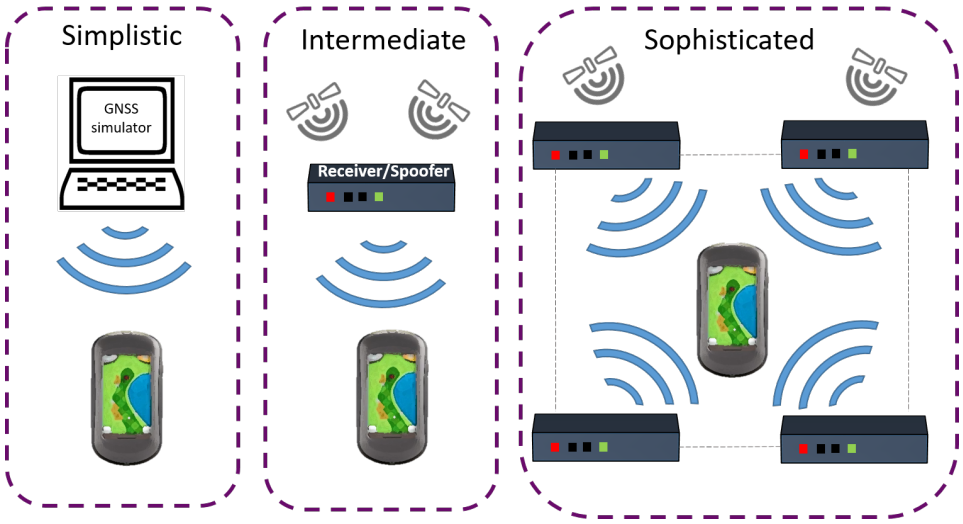

Fig. 2.1 Illustration of the classification of spoofing attacks, simplistic, intermediate and sophisticated. We can observed how the simplistic attack uses a GNSS simulator, while the intermediate attack is performed by a receiver/spoofer, able to generate signals aligned with the contellation. The sophisticated attack consists on several receiver/spoofer synchronized and transmitting at the same time from different locations. *Inspired by illustration in* [23]

signals in order to force the receiver to re-acquire so the receiver would lock in to the simulated signal from the spoofer.

2.4.2 Intermediate spoofing attacks

The second type of attack is known as intermediate attack or receiver-based spoofing attack. This type of spoofer has a built-in receiver that collects and tracks the satellite signal parameters, in order to generate a new signal that is consistent with the current constellation and transmit it to the target receiver [36]. A graphical representation is shown in the center of Fig. 2.1. This type of spoofing attacks are the focus of this research, given that its feasibility has been proved, and that it is usually able to modify the target's position solution without raising warnings or creating discontinuities in the PVT solution.

2.4.3 Sophisticated spoofing attack

The third type is called the sophisticated receiver-based spoofer and it can be seen to the right of Fig. 2.1. It aims to overcome one weakness of the intermediate attack, which is that it only broadcasts from a single antenna and direction. The sophisticated version uses several different antennas to broadcast each satellite signal in order to be undetectable through anti-spoofing techniques that rely on angle of arrival discrimination. However, these attacks have a much higher complexity level, given the synchronization and communication process between each individual transmitter, making it very difficult to realize and not suitable for real scenario examples.

2.5 Review of anti-spoofing techniques literature

Since the first major publication reporting the feasibility of building a spoofer with a software defined receiver and low cost components [36], the GNSS community have been continuously developing different anti-spoofing techniques, aimed at defending against the malicious signals at different stages of the receiver. Generally, the main goal is to detect the attack and raise a warning in case of unreliability of the position solution. Additionally some of them can also perform mitigation of spoofing effects. In this Section we present a general review of anti-spoofing techniques.

Given the increasing concern over spoofing attacks, the first COTS receiver that includes spoofing detection as a feature has recently been released bu Ublox [29]. Even if this is a positive step forward in spoofing detection, it still represents a very small part of the industry and to the best of our knowledge, no other receiver provides warnings against such attacks. Nevertheless, there is no public information on what type of anti-spoofing technique the Ublox receiver has inside, thus making it difficult to trust.

Different classifications have been proposed in literature. For example, [42, 49] divide the different techniques in two big groups, the spoofing *detection* and spoofing *mitigation* techniques. This classification allows for a distinction between goals of the techniques. Spoofing detection algorithms concentrate on discriminating spoofing scenarios, but not necessarily perform countermeasures against the attack, on the other hand mitigation techniques focus on reducing the effects of the spoofing attacks, with the goal of computing a reliable navigation solution [42].

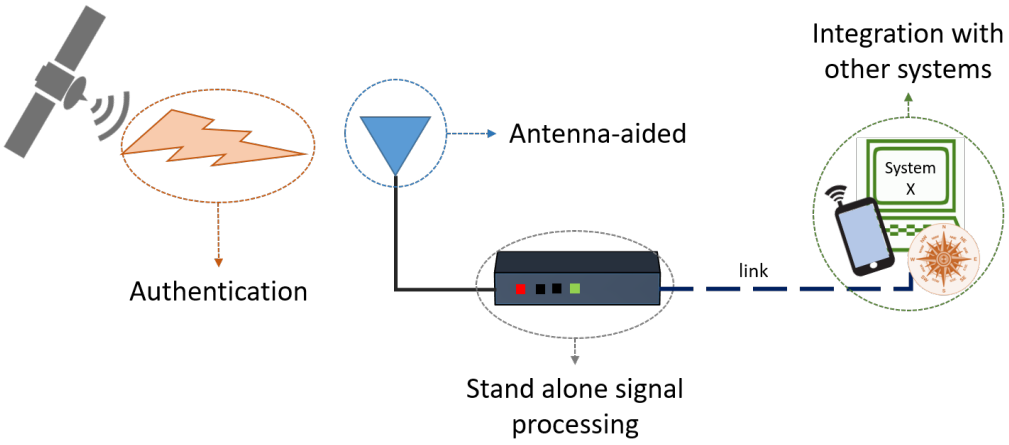

Fig. 2.2 Graphical representaion of the four main groups of the anti-spoofing techniques, as classified in this Chapter and the main focus of each of them.

Another possible nano-classification is the one done in [82, 83], where they divide antispoofing techniques in *cryptographic* and *non cryptographic*. This classification is useful if we are to only consider cryptrogaphic solutions for GNSS anti-spoofing, which is not the case of this thesis.

For this Chapter, a classification approach as the one presented in [23] is decided to be used, which is focused on the elements used for spoofing detection. This classification provides four big groups, detailed in Section 2.5.1. Using this classification we can group together all the techniques studied in this thesis in a single group, known as *stand alone signal processing techniques*.

2.5.1 Classification of anti-spoofing techniques

Anti-spoofing techniques have very different approaches to the spoofing detection problem, at least for what concerns civil receivers, they can be classified into four main groups as observed in Fig. 2.2

In Fig. 2.2, we can observe the main focus of each type of technique. From left to right of the Figure we observe, first the *authentication* techniques which focus on characteristics of the signal coming from the satellite and distinguishing if it is fake or not. Then *antenna-aided* techniques, use specialized antenna arrays for distinguishing between a satellite signal coming from the sky and the spoofing signal

coming at low elevations. Afterwards, *stand alone signal processing* techniques are implemented within the receiver, and use signal processing for discriminating the spoofing signal. Finally, *integration with other systems* provides protection from spoofing attacks through cross-checks done with other independent systems. In following Sections, each group is explored.

Authentication techniques

A definition for GNSS authentication is given in [42], as the "certification that a received signal is not counterfeit, that it generates from a GNSS satellite and not a spoofer".

Utilizing authentication techniques is a cryptographic countermeassure to spoofing attacks. These types of techniques are based on the encryption of the digital sign portions of the broadcast GNSS signals. As detailed in [23], different authentication solutions aim at protecting two important elements of the GNSS signals: the spreading code chips and the navigation message data bits. To do this, there are several approaches that can be followed:

- Introducing features able to make the signal difficult to be generated by the attacker [27, 20].
- Authentication of the satellite signal, by comparing outputs among different receivers or among different frequencies (e.g., L1/L2) [52]
- Exploit the transmission of restricted-access GNSS signals (e.g., the military GPS L1 P(Y) code), and cross-compare signals received at different locations [33, 20, 88]
- Position authentication of civil receivers, based on a client-server approach [74]
- Provide an open and encrypted service on the GNSS system [69].

In general, all these techniques rely on the modification of the current civil SIS or at least the navigation message content. Alternatively they rely on the presence of other receivers, signal and/or servers, thus making their overall usage limited and their realization difficult.

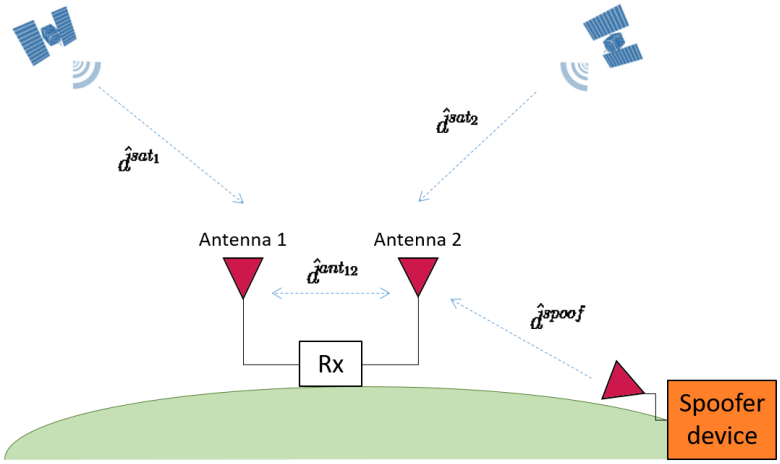

Fig. 2.3 Antenna-aided technique example. In this scenario, multiple antennas connected with each other, are used to discriminate the angle of arrival of the spoofing signal

Antenna-aided techniques

These techniques exploit the advantages of an antenna array and are based on the wide spatial correlation that the satellite signal has with respect to the spoofer signal during an intermediate spoofing attack. These techniques are vastly different and they can be further divided in:

- Angle of Arrival detection with multiple antennas [64, 60]
- Use of synthetic arrays for angle of arrival discrimination [62]
- Beam forming techniques that are able to mitigate the attacks by pointing null lobes at the spoofing signal [19, 56]
- Specialized single antennas for aircraft applications [57, 58]
- Correlation of carrier phase with antenna motion [71]

In Fig. 2.3 we show an example of a multiple antenna array, in which they are used to discriminate the angle of arrival of the spoofing signal. As can be seen, the vector of arrival of the spoofer signal $\hat{\mathbf{d}}^{spoof}$, will be distinguishable from the satellite ones. Among the different anti-spoofing techniques, antenna-aided techniques are considered the most powerful, given that most of them are able

not only to detect the spoofer, but also mitigate its effects through null lobbing. Additionally these technique are very reliable for detection of intermediate spoofing attacks. Nevertheless, these techniques require specialized hardware at the receiver level, very detailed calibration and are likely expensive.

Stand alone signal processing techniques

There is an increasing interest in stand-alone techniques, that can be implemented by a receiver and provide spoofing attack discrimination. This interest is driven mainly by the fact that modernized GPS does not foresee an implementation of civil authenticated signals in the near future [23] and that antenna configurations require additional external and specialized hardware to function. These factors create an opening for techniques that are contained within the receiver, that can distinguish between the current civil signals and spoofing signals and that do not require external hardware or communication. These represent the main characteristics of stand alone signal processing techniques as envisioned in this Section.

These methods are based on signal processing techniques, which mainly focus on aspects that differ between the spoofing and the satellite signal. Many different techniques have been proposed in this category, such as:

- Signal power monitoring, either monitoring the C/N_0 [41], the absolute in-channel power measurements using AGC [3] or relative power variations [43].
- Analysis of the distribution of the correlator output [85].
- Signal quality monitoring, which aims at detecting distortions on the correlation function via the use of ratio metric tests. [67, 81, 84].
- Time of Arrival and other consistency checks of the navigation bits [37].
- Goodness-of-fit tests [21].
- Receiver Autonomous Integrity Monitoring (RAIM) discrimination algorithms [43, 48]
- Split estimation of the received signal components, using multiple DLLs for anti-spoofing [24, 4].

All of these techniques are contained within the GNSS receiver and do not require external elements. The focus of this thesis is on stand alone signal processing techniques, and several of the aforementioned techniques are revisited. Also we developed detection algorithms for each one, widening their usage and improving their detection capabilities. We also validate their performance and propose methods to integrate them.

Integration with external systems

There exist several other techniques that focus on the integration with external systems, outside of GNSS. If the spoofing attack is affecting the GNSS signal, data provided by other systems can be used to cross-check the GNSS measurements [23]. These techniques include the *integration with inertial sensors* and the *integration with communication systems*, among others.

Inertial sensors provide a natural barrier to the spoofing presence because they are able to generate their own position solution, thus enabling an easy cross-check between the two solutions [42, 49]. One advantage of this technique is that the fusion between inertial systems and GNSS is a common practice to improve each system's accuracy. Both systems are complimentary and the redundancy that they present leads to major advantages in PVT accuracy.

Integration with communication systems enables the cross-check of positions obtained from cellphone tower or Wi-fi communications. These other communication systems can provide a rough position, but are enough to check for consistency between the two solutions [42]. Additionally, messages can be transmitted through a secure communication channel in order to obtain approximated positions or aids in the position computation, as done frequently in smartphones.

2.6 Detection and Mitigation of spoofing attacks

This thesis focus on different aspects of anti-spoofing techniques. As stated before, some of the techniques are designed to do spoofing detection and raise alarms when an attack is going on, but others, not only are able to detect the spoofer but also are able to mitigate its effects. The distinction between mitigation and detection is important, in order to understand the limits of the analyzed anti-spoofing techniques.

For example, in [42] an analysis on the different techniques belonging to each category is presented and interesting graphs are shown. Another in depth review of detection and mitigation comparison, and the provided protection against different configurations of the spoofer was done in [70].

In this thesis, the SQM is used as a main focus for detection strategies. It has the advantage of having a low detection latency and very simple detection approach. It is improved by additional checks, such as observations of the AGC gain, check over time consistency and number of satellites affected, in order to lower its false alarm and miss detection probabilities.

For mitigation techniques, two very different approaches are proposed in this work. Each technique detects the spoofing events by its own means, but the general mitigation is done by estimating the separation between the satellite and the spoofer signal and obtaining the pseudoranges using the satellite signals, thus effectively reducing the spoofer effects.

It is important to mention that not every application is suitable for mitigation because the accuracy may not meet specific requirements. A mitigated PVT solution may not be as accurate as the one obtained by means of the clean satellite signals, due to the difficulty of completely removing the spoofer signal from the receiver. Nevertheless, mitigation is a powerful feature because the receiver can keep computing PVT solutions that are not controlled by the spoofing signal, especially if the application doesn't require sub-meter levels of accuracy, like geofencing or road tolling.

Part II

Spoofing Detection

Chapter 3

Signal Quality Monitoring

In this Chapter we introduce the general working procedure of SQM techniques. These techniques are used for detection of distortions in the correlation function, and these distortions are characteristics of the presence of impairment signals in the receivers. These additional signals may be introduced by multipath effects or by interference presence. SQM techniques will be a fundamental part of this thesis, since they are used, in combination with other techniques, to improve spoofing detection capabilities.

For example, in Chapter 4 the SQM will be combined with other observables of the GNSS receiver in order to increase the detection capabilities and distinguish between multipath and spoofing attacks. In Chapter 5 the SQM will be combined with observations of the AGC measurements in order to detect overpowered spoofing attacks. Finally, the SQM will be used along the ECADLL in order to improve spoofing detection delay and to aid the monitoring block turn On/Off the different Units in Chapter 7.

The SQM is a powerful and proven technique that was originally introduced in [65] to detect multipath signals and it was later used, in different versions, for spoofing detection [76, 67, 81].

The work presented in this thesis is based on [5, 54].

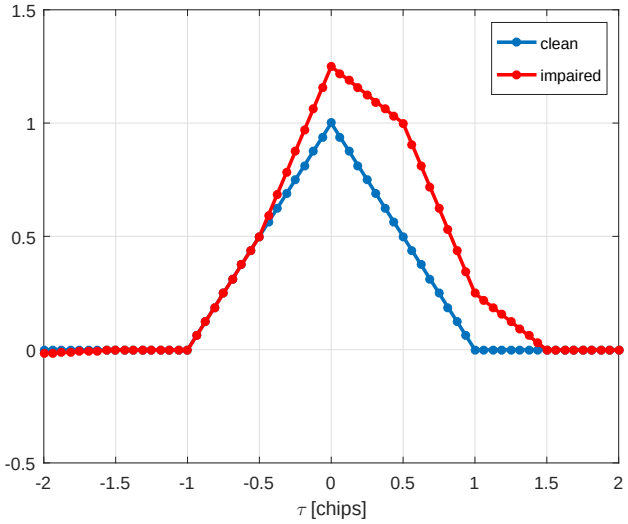

Fig. 3.1 Correlation function example for GPS L1 C/A code, in clean and impaired scenarios. Each pair of dots can indicate a correlator pair, used to track the shape of the function

3.1 Basic concepts

In recent literature several SQM techniques have been discussed and tested. It is a simple approach that can be used to detect and raise alarms, in case of distortions in the correlation function, where the metrics do not pass a certain quality level [23].

SQM techniques are based on the observation, by means of correlator pairs, of the correlation function between the satellite signal and the local code replica. A correlator pair indicates two correlators, symmetrically placed around the prompt correlator, with the same absolute delay, but with different sign

In Fig. 3.1, we can observe an example of a correlation function for the GPS L1 C/A code. The curve in blue is the correlation function in absence of impairments, when it has its theoretical triangular shape. In red, we can observe how the presence of additional signals modifies the function shape. By means of correlators pairs, the SQM is able to keep track of the asymmetries in the shape. Generally, the metrics used for the detection are constructed using a linear combination of the values obtained by correlators pairs. In this Chapter, only one pair of correlators will be used, meaning that a total of two correlators values are used, one to obtain an early value of the correlation and one for the late value.

Different metrics have been proposed using combinations of correlators and are detailed in [66, 81, 49]. Two of the most common test metrics are the *delta test* and the *ratio test* [81, 54, 67], and are the two that will be used in this thesis.

The correlator values for the in-phase branch of a coherent DLL will be denoted as E_m, L_m and P. These represent the early (E), late (L) and prompt (P) correlator outputs with a spacing of m chips w.r.t. the prompt location and with $0 < m < 1$. In a non-coherent DLL, these correlators will be the square sum of the in-phase and quadrature correlators. Throughout this thesis we work under the assumption of a locked PLL, and thus we use a coherent DLL.

3.1.1 The delta test

The *delta test* is used to detect asymmetries of the correlation function, by means of the difference between the two values in a correlators pair. This can be defined, for epoch k, as:

$$\Delta[k] = \frac{E_m[k] - L_m[k]}{P[k]} \tag{3.1}$$

and the division by the prompt value is used to normalize the metric, making it independent of the amplitude of the received signal. The *delta test* metric will be used in Chapter 5 in order to analyze the correlation function shape of a commercial receiver. If we analyze the metric, it is easy to see that the mean value of Δ will tend to zero in a clean data set, and in case of asymmetries, Δ will be a positive or negative number, based on the delay and phase of the interference signal.

3.1.2 The ratio test

The *ratio test* is used to detect asymmetries of the correlation function by observing the relationship between the sum of early and late correlators, w.r.t. the prompt value. This metric can be defined for epoch k as:

$$M[k] = \frac{E_m[k] + L_m[k]}{\xi_m P[k]} \tag{3.2}$$

where ξ_m is a constant factor, that represents the slope of the correlation function. For example, for the GPS C/A code and m equal to the chip duration, $\xi = 1$. The

ratio test will be used as the main metric for Chapters 3 and 4, and the resulting metric will be referred to as Ratio Metric (RM).

Assuming to work with a locked PLL and the in-phase branch of a coherent DLL, E_m, L_m and P can be modelled as identically distributed Gaussian random variables. In fact, in the integration process, the independent white noise samples of the received signal, generate outputs, whose probability density function (pdf) is Gaussian [40]. $M[k]$ is the ratio between two Gaussian random processes, the summation $E_m[k]+L_m[k]$ and the value of $P[k]$, and generally, it is no longer Gaussian. When the spacing m is large the variables can be considered independent, while for narrow spacing their correlation cannot be neglected

However, if the noise at the output of the prompt correlator is negligible, $P[k]$ can be approximated as a known constant, whose value mainly depends on the signal power. This approximation is quite realistic when the receiver works with high values of C/N_0 [40, 39], specifically for signals with C/N_0 values greater than 46 dBHz as can be seen in Fig. 3.2. In the Figure we observe the quantile-quantile plot [87], comparing simulated samples with the theoretical normal distribution of different RM, resulting from GPS signals with the indicated value of C/N_0. The hypothesis of Gaussian distribution was also tested by means of the Lilliefors hypothesis test for the goodness of fit to a normal distribution. The Matlab function *lillietest* was used for this test and it delivered $h = 0$ for the signals with $C/N-0$ above 46 dBHz and $h = 1$ for the rest, meaning that for the former signals, the hypothesis of Gaussian distribution cannot be rejected within the 5% significance level [51].

Under this assumption, the metric $M[k]$ can be written as

$$M[k] = \mu[k] + N[k] \tag{3.3}$$

where $\mu[k]$ is the mean value due to the signal component, and $N[k]$ is a zero mean iid Gaussian process with known variance σ^2, due to the noise component. The value of σ depends on the signal power, DLL spacing, and shape of the correlation function, that can be directly evaluated by the receiver. In the rest of the thesis, the general probability density function (pdf) of M will be referred to as $f_{\mathrm{M}}(\alpha)$ and the evaluation of the overall probabilities will be done under the hypothesis of high values of C/N_0. Under this assumption M can be approximated as normally distributed, i.e. $M \sim N(\mu, \sigma^2)$, with mean value equal to μ and variance σ^2. The

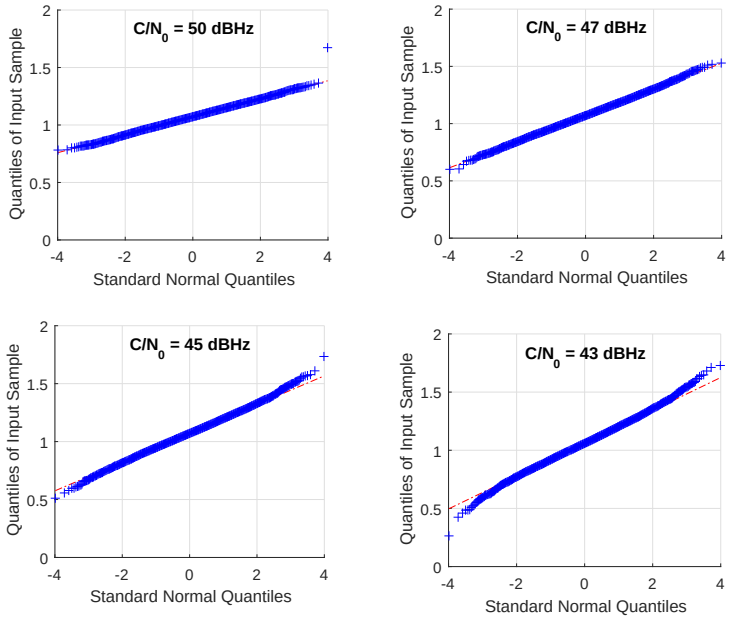

Fig. 3.2 Comparison of different quantile-quantile plots for Ratio metric resulting from signals with different values of C/N_0

pdf of M is written as:

$$f_{\mathrm{M}}(\alpha) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\alpha-\mu)^2}{2\sigma}} \tag{3.4}$$

3.2 Mathematical derivation

A Neyman-Pearson (NP) detector [46] is used to take the decision between two hypothesis, H_0 and H_1, defined as

$$\begin{aligned} &H_0 : \text{the } \textit{null hypothesis}\text{, absence of disturbances} \\ &H_1 : \text{the } \textit{alternative hypothesis}\text{, presence of disturbances} \end{aligned} \tag{3.5}$$

Distortions in the correlation function will modify the mean value of the metric M because the correlator values will change, as observed in Fig. 3.1. Consequently, M

will have a different distribution under each hypothesis, such as:

$$\begin{aligned} H_0 &: M \sim N(\mu_0, \sigma^2) \\ H_1 &: M \sim N(\mu_1, \sigma^2) \end{aligned} \tag{3.6}$$

where the value of the variance, σ^2, is assumed the same between both hypothesis because it only depends on the noise level of the correlators and not on the expected values. The NP theorem assures to have the maximum probability of detection P_D, given a fixed probability of false alarm P_{FA}, and, as detailed in [46], a NP decision strategy is based on the definition of a likelihood ratio (LR) test to be compared against a threshold γ_L:

$$LR(M[k]) = \frac{p(M[k];H_1)}{p(M[k];H_0)} > \gamma_L \tag{3.7}$$

from which a test applied to a set of observables can be derived. By omitting the dependence on the epoch k, from (3.7), as derived in Appendix B, the LR test can be expressed as:

$$M > \frac{\sigma^2 ln\gamma_L}{\mu_1 - \mu_0} + \frac{\mu_1 + \mu_0}{2} = \gamma \tag{3.8}$$

where μ_0 and μ_1 are the values of μ in the absence of noise when H_0 and H_1 are respectively verified. Eq. (3.8) is valid for $\Delta\mu = \mu_1 - \mu_0 > 0$ and a similar expression can be found for $\Delta\mu < 0$. We can observe that the threshold γ depends on γ_L and could be obtained from the P_{FA} of the likelihood test LR, in (3.7).

In practice, in order to obtain γ when M is assumed Gaussian, it can be directly computed from the desired false alarm probability of M, $P_{\text{FA},M}$, as:

$$\gamma = \sqrt{2}\sigma \cdot \text{erfc}^{-1}(2P_{\text{FA},M}) + \mu_0 \tag{3.9}$$

As can be observed in 3.9, threshold γ is only based on $p(M[k];H_0)$ and can be obtained by using the nominal statistics of the metric under hypothesis H_0.

The decision is taken for H_0 if the metric is below the threshold γ, and for H_1 otherwise:

$$M \begin{cases} < \gamma \longrightarrow H_0 \\ \geq \gamma \longrightarrow H_1 \end{cases} \tag{3.10}$$

It is worth noticing that the threshold γ tends to diverge towards infinite as μ_1 approaches μ_0. This relates on the fact that the ratio metric is effective only if the two values μ_1 and μ_0 are well distinct, since they have to be used to discriminate between H_0 and H_1. If $\mu_1 \simeq \mu_0$ no discrimination is possible and the test will not detect the spoofer presence. Nevertheless, in case of asymmetries in the correlation function, the value of μ_1 will increase rapidly with the increase of $\Delta\tau$ and the amplitude w.r.t. the satellite signal. In case of spoofing attack, we will assume that the spoofing signal is in phase with the satellite signal, thus making M, and consequently μ_1, increase.

3.3 Spoofing detection based on the SQM

The equations presented in Section 3.2 describe possible tools to detect the presence of a spoofing attack by observing asymmetries in the correlation function, assuming that $f_{\mathrm{M}}(m|H_1) = N(\mu_1, \sigma^2)$. This assumption is quite realistic as the mean value of the metric, μ_1, will be modified by the presence of additional signals in the correlation space and the Gaussian distribution will be maintained for high C/N_0.

As stated before, the SQM was originally developed for multipath detection, and it was later proposed for spoofing detection. In this Section, we analyze the SQM as a spoofing detection technique and develop a detection algorithm, based on the SQM, able to take continuous decisions on the spoofing presence. The detection algorithm is based on the work done in [5, 54], and it takes advantage of the fact that the spoofing attack is a continuous phenomenon.

As detailed in Chapter 2, a spoofer needs a continuous signal, present inside the correlation space, in order to slowly modify the delay, thus affecting the PVT solution. If the spoofing signal is only present for a very short time, the DLL will keep its lock on the satellite signal. This temporal continuity is a key feature, exploited in the spoofing detection algorithm.

This temporal continuity can be exploited by observing a limited set of samples inside a Detection Window (DW). During one DW, the receiver will collect outputs of M and will be able to take the decisions based on a set of samples and not only on one value of metric M. This method allows for a decision once every DW, and it will help in reducing the effects of false alarms. A single event, where metric $M[k] > \gamma$

for only one value of k, is probably not due to the spoofing presence, but to the noise component.

Inside a receiver, in order to take the decision, different steps needs to be followed in order to complete the detection process. The proposed algorithm can be divided into 4 steps:

1. As an initial step, the algorithm needs to obtain the values of μ_0 and σ. These values can be obtained from the receiver parameters, knowing that the theoretical mean value of the metric is 1, given by the sum $(E_m + L_m)/\xi_m \approx P$, and using (12.6) of [59], the variance of each correlator can be computed as:

$$\sigma = \frac{N_0}{2T_{CO}} \tag{3.11}$$

   where N_0 is the noise density in the receiver and T_{CO} is the coherent integration time. Another way to obtain statistics for the correlators is with controlled environment simulations and data collections using the same receiver configuration that will be used later for spoofing detection. Another possible approach could be the use of a calibration phase, computing the parameters as soon as the receiver is turned on, assuming it will be free of spoofing presence.

2. As a second step, the algorithm computes the threshold γ for the spoofing detection using (3.9).

3. Third, the algorithm will collect a set of samples of the metric M, inside a DW. The number of samples observed will depend on the duration of the DW and on the frequency at which M values are obtained. In the case of a software receiver, this frequency is usually equal to the inverse of the integration time. In commercial receivers, the output rate of the correlator output depends on the manufacturer.

4. Finally, the algorithm takes the decision based on the percentage $x\%$ of samples that are above the threshold. In this way, the algorithm effectively excludes instantaneous and spurious events. This decision can be expressed as:

$$\begin{cases} H_1 \text{ if } M \geq \gamma \text{ for } x\% \text{ of the DW} \\ H_0 \text{ otherwise} \end{cases} \tag{3.12}$$

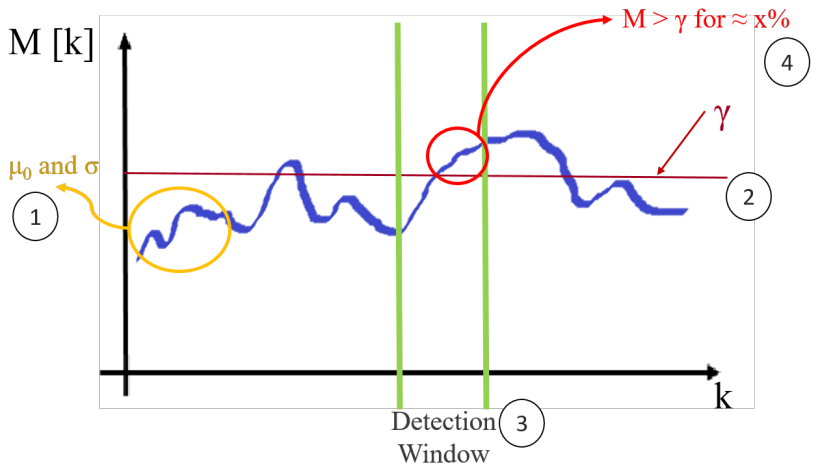

Fig. 3.3 Graphical example of the steps taken by the detection algorithm

In our case, for the software receiver implementation used for this Chapter, based on [13], we compute values for μ_0 and σ every time the receiver is started and a new test is going to be realized. In this way, we obtain statistics for clean environment related to the current scenario and configuration. This means that a *calibration phase* is performed once the test is started, and usually last for 10 seconds. During these 10 seconds the assumption is that no spoofing signal is present and thus we are able to obtain statistics for $f_{\mathrm{M}}(\alpha)$.

In Fig. 3.3 we can observe a graphical example of the procedure taken by the algorithm and in Fig. 3.4 a flow chart of the algorithm is presented.

Considering the observation window to make a decision, we are able to detect asymmetries in the correlation function and warn the receiver of their presence every DW. One drawback is that asymmetries generated due to multipath signal will also be flagged as present, so another layer of discrimination should be added on top in order to distinguish between the two effects. This situation is addressed in Chapter 4, where an evolved metric is proposed for the discrimination.

3.4 Results using SQM for spoofing detection

In this Section we present the results obtained with the spoofing detection algorithm presented in Section 3.3 by means of the TEXBAT datasets. The datasets are

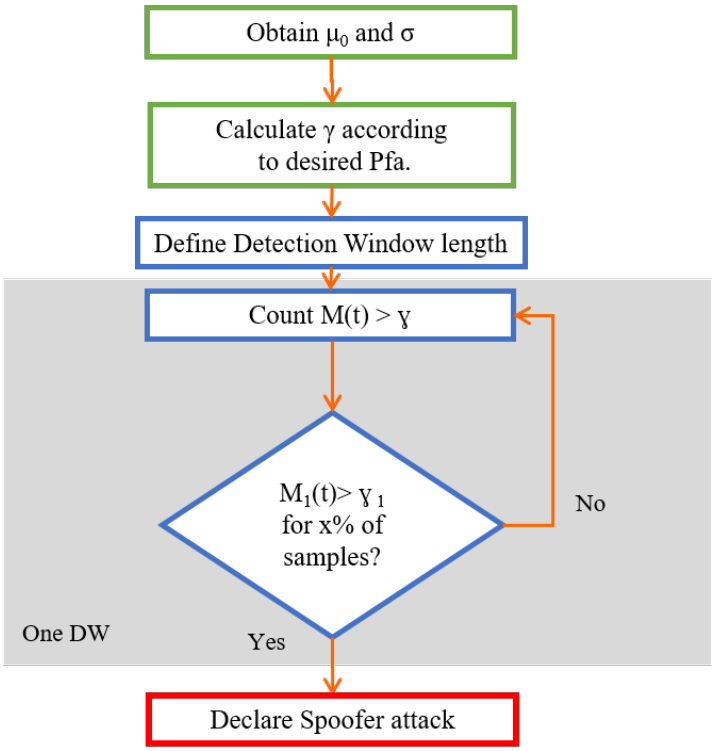

Fig. 3.4 Flow chart of the steps taken by the spoofing detection algorithm

Table 3.1 Parameters for spoofing detection using SQM

Parameter	Value
x	50%
DW duration	1 s
m	0.5 chip
$P_{\text{fa},M}$	10^{-2}

described in Appendix A.1. From the different datasets available in the battery, in this Section we use scenarios ds4 and ds6, and Static Clean as a reference, for validation of the algorithm. These scenarios allows for validation versus matched-power static (ds4) and dynamic (ds6) spoofing attacks.

The parameters used for obtaining these results are detailed in Table 3.1

Clean static scenario

In this Section we present results for the processing of the Clean Static scenario of the TEXBAT. It is used as a baseline of comparison for the other spoofed scenarios. It was recorded in an open sky static environment, so no false alarms are expected.

In Fig. 3.5 we can observe an example on how the metric M behaves in time for PRN number 13. As we can see the behavior is very stable through the whole test and no false alarms are detected as expected. In Fig. 3.6, the decision taken for all tracked satellites in this scenario are shown. We observe how none of the satellites triggers the detector in 3.12, so no false alarms are declared. With these results we have a reference for the behavior of the metric, and the spoofing detection algorithm, for scenarios without the presence of additional signal and in clean and high C/N_0 environment.

Static scenario ds4

In this Section, we present the results for scenario ds4 of the TEXBAT. It is a static scenario, where the spoofing attack affects the position solution of the receiver. This means that the spoofer will modify each single satellite with a unique delay, in order to obtain the desired modification of the position. This feature can be observed in Fig. 3.7 where the metric M is shown for two different satellites, PRN numbers 3 and

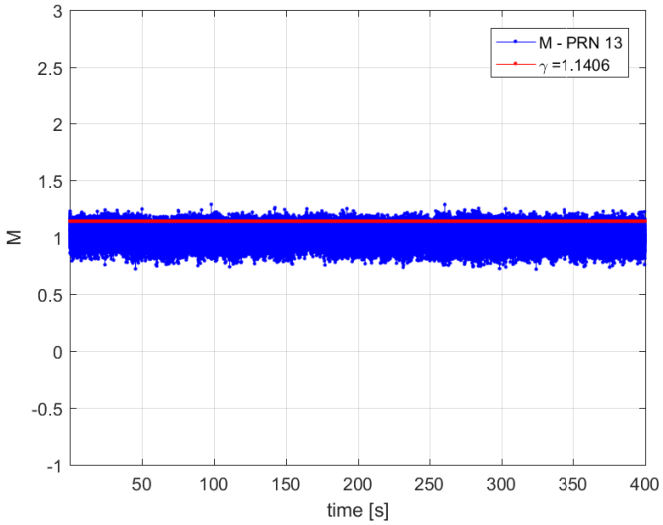

Fig. 3.5 Behavior of the metric M for the Clean Static dataset of the TEXBAT, along with the threshold γ computed during the calibration phase. PRN number 13 is shown

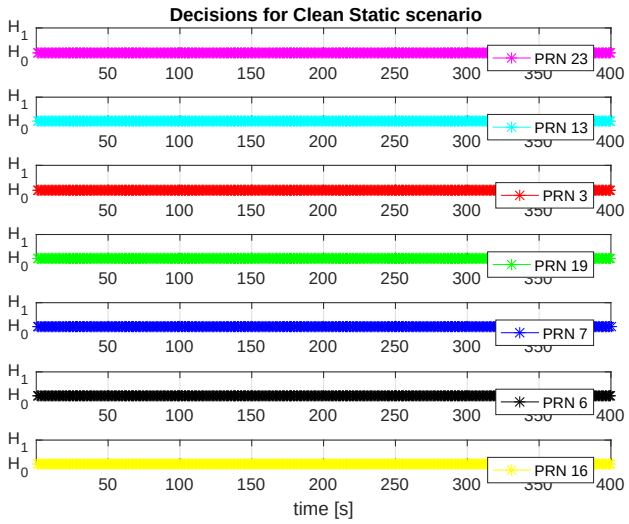

Fig. 3.6 Decision taken for each satellite of the Clean Static dataset of the TEXBAT

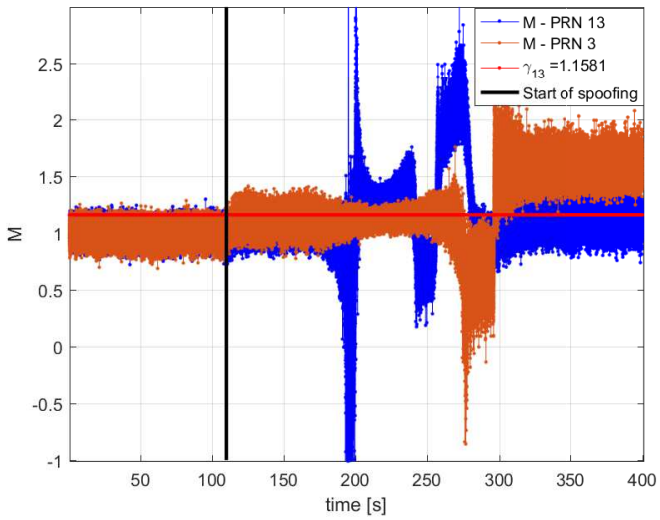

Fig. 3.7 Behavior of the metric M for scenario ds4 of TEXBAT. PRN 13 (in blue) and 3 (in orange) are shown. The threshold γ computed for PRN 13 is also shown as reference. The spoofing attack starts at time instant 110 seconds. We observe how the Metric behaves differently for each satellite due to the nature of the attack.

13, under the effects of the spoofing signal. We can observe how the two trends are different and effects on the metric are observed over the duration of the attack, from 110 s to the end of the test. In Fig. 3.8, the decision taken for all the tracked satellites is presented along with the 3D rms error introduced by the spoofing signals. We can observe how the discrimination for spoofing presence is done for all satellites, at different time instants, depending on the specific delay introduced by the spoofing in each of them. This delay will affect M in a different way as observed in Fig. 3.7. We can also observed the total 3D error is in the order of a several hundred meters, and that the detection of the spoofing attack is observed in more satellites as the total 3D error increases. With this example we validate the spoofing detection capabilities of the algorithm in a static scenario with a position push. All tracked satellites are detected as spoofed and no false alarms are detected while the spoofer is not present, during the first 100 seconds of the data. One limitation of the technique is that the spoofing attack is not detected for the whole duration of the attack, but only during a limited time window for each satellite, making it a transient indicator of the spoofer presence. It is important to recall that for spoofing detection algorithms, the receiver will be warned once a single spoofing is detected and it should not trust the PVT solution obtained after the warning.

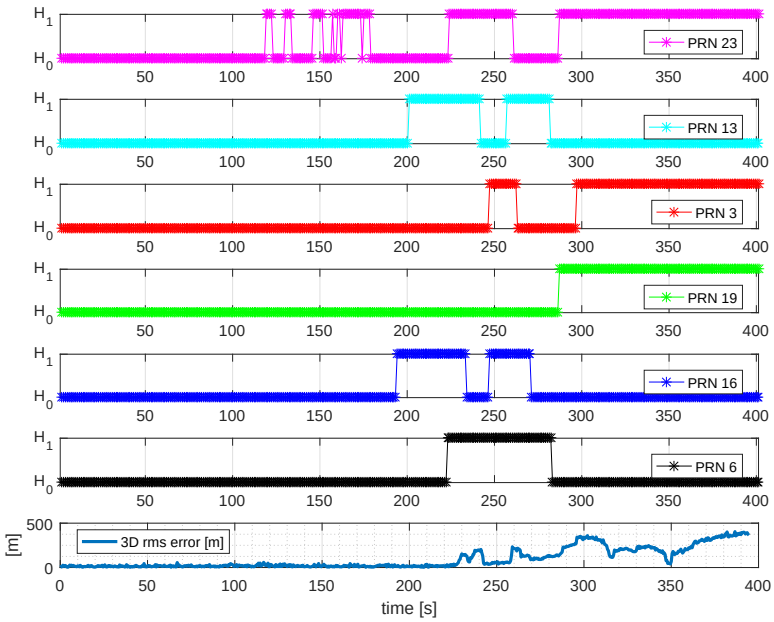

Fig. 3.8 Decision taken for each satellite of TEXBAT ds4 along with the 3D rms error introduced by the spoofer. All satellites are declared as spoofed for a period of time of the data

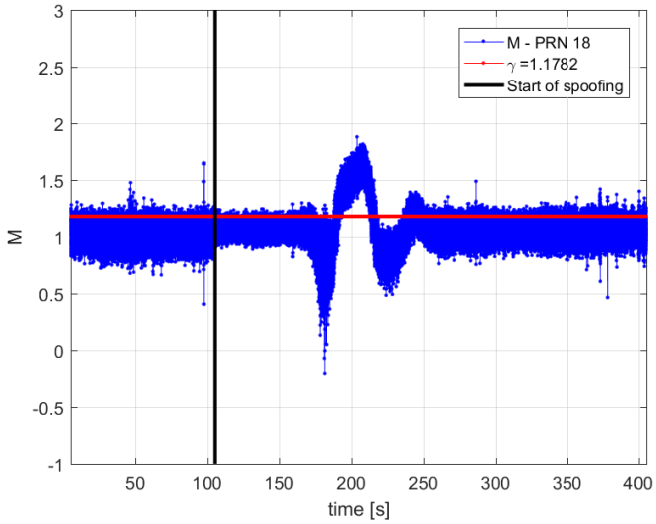

Fig. 3.9 Behavior of the metric M for scenario ds6 of TEXBAT. PRN 18 (in blue) is shown along with the threshold γ (in red). In black the start of the spoofing attack is indicated, at around 105 seconds. The beginning of the attacks does not mean that the delay of the signal is being, it only means that the spoofing signal is present.

Dynamic scenario ds6

Finally we present the results for the detection of TEXBAT scenario ds6. This is a dynamic scenario where a matched-power spoofing signal modifies the position of the receiver. In Fig. 3.9 we observe an example of the trend of M, obtained for one of the satellites, PRN 18. We observe a similar trend to the one observed for ds4, where the metric is affect during a specific time window. The decisions taken for the different satellites are shown in Fig. 3.10 and we can see similar performance to the one observed for ds4. Each satellite is affected in a different fashion and for a limited window. Additionally, in this scenario, satellite signal with PRN 9 is never declared as spoofed, resulting in a missed detection. We can observe in this dataset how large errors in the position, up to 700 meters in the three dimensions, are introduced by the spoofing signal. Additionally, when the spoofing presence is detected, errors of 300 meters are already being introduced in the position.

In this Section we validated the spoofing detection algorithm against a set of datasets and the results demonstrate the potential for spoofing detection. It is clear that the spoofing detection capabilities of the technique are dependent on different parameters of the algorithm. These parameters need to be explored further and

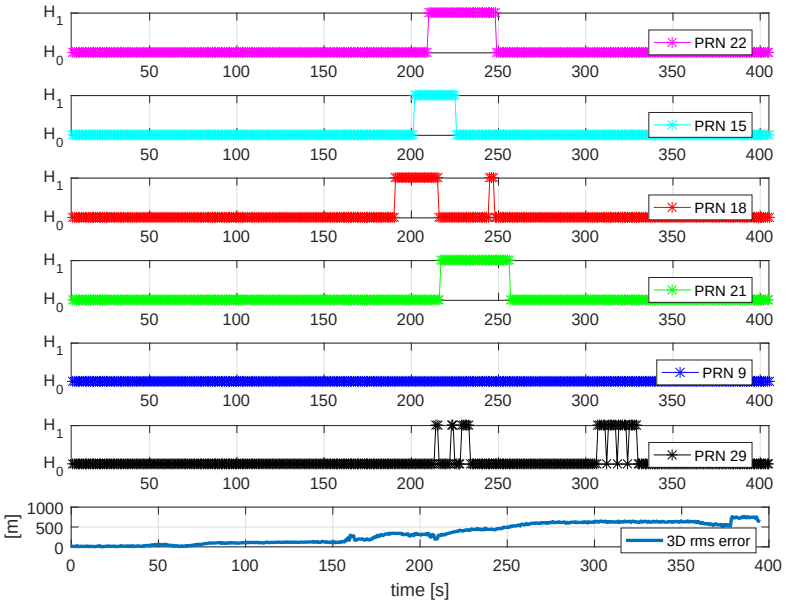

Fig. 3.10 Decision taken for each satellite of TEXBAT ds6 and the total 3D rms error introduced by the spoofer. Five out of six satellites are declared as spoofed and each one is affected in a period of time of the data

chosen according to the application. The values of the duration of the DW, the early late spacing m and the percentage $x\%$ to choose for declaration of spoofing presence will be further explored in Chapter 4, also considering the effects of multipath signals.

Introducing a second threshold for cases reducing the value of the metric

Observing Figs. 3.7 and 3.9, we can see that the distortions in the correlation function may lower the mean value of the metric, instead of raising it. Normally, when the spoofing signal is aligned in phase w.r.t. the satellite signal, the effects of the spoofing presence generate a positive bias in the metric M.

Given that this is not always the case, the algorithm may consider the introduction of a second threshold in order to detect a wider range of spoofing attacks. The threshold is computed using (3.9), but subtracting its value from μ_0, and the detection algorithm works in the same way as before, but only considering samples that are below the threshold.

With the introduction of the second threshold, the detection capabilities are improved as can be seen by comparing Figs. 3.8 and 3.11. The improvement of the detection on each channel is between 5 and 10 %.

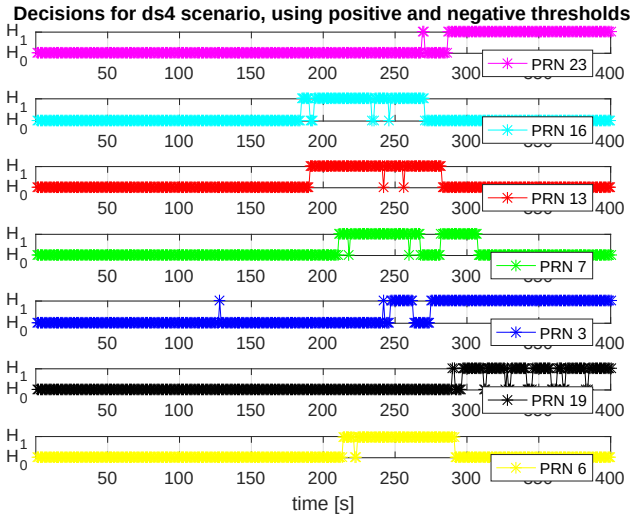

Fig. 3.11 Decision taken for each satellite of TEXBAT ds4 using both upper and lower thresholds. All satellites are declared as spoofed for a period of time of the data, and we can observe some improvement in the detection w.r.t. Fig. 3.8

Chapter 4

Improvements to SQM for discrimination between multipath and spoofing events

In general, SQM techniques have been previously employed to monitor the correlation peak quality in multipath environments [65]. Since the distortions in the correlation peak, caused by multipath events, might be similar to those generated by a spoofing attack [76], the technique can also be used for spoofing detection. In order to discriminate between these two phenomena, this Chapter analyses different elements that characterize a spoofing attack and proposes a new metric, able to contemporaneously detect the disturbance and discern its nature.

The objective of this Chapter is to present a new *multidimensional ratio metric*, indicated as β, present a statistical analysis of it, and validate its performance in scenarios affected by either spoofing or multipath components. The new metric β, is based on M as defined in Chapter 3, and is able to discern whether the correlation distortions are due to spoofing signals or multipath components. In fact, even though the effects on the correlation function might be similar, the two phenomena present unique characteristics, as explained hereafter.

This Chapter is based on the work presented in [55].

4.1 Preliminary example

In this initial Section, we highlight the similarities and differences between a multipath heavy scenario and a spoofing attack scenario.

In Figs. 4.1 and Fig. 4.2 we show the behavior of the ratio metric M applied at two different datasets, which characteristics are detailed in Table 4.1. Both of them are urban road scenarios, in which the data was recorded from an antenna mounted on top of a car. As for the spoofed test environment, the TEXBAT ds6 dataset, described in Appendix A.1 and in [38], was used. As for the multipath, a real dataset recorded in urban road environments of Torino, Italy, described in Appendix A.3 was used to highlight the different scenario characteristics. In the case of ds6, a spoofing signal 0.8 dB stronger than the genuine one (matched-power) is added to the genuine signal, causing the distortion of the position solution. The dataset To-1 was collected in dense urban environment, where the presence of multipath components is very likely.

Table 4.1 Scenarios used for the preliminary example

Name	Place	Date	Description	Speed
ds6	Austin, Texas	01/2011	Dynamic matched-power position push	40 km/h
To-1	Turin, Italy	09/2013	Dynamic urban	50 km/h

In the top plots of Fig. 4.1 and Fig. 4.2 the trend of M is plotted along with the threshold γ. The decision on the hypothesis is also shown (bottom plot). The decision is taken following the algorithm explained in 3.3, using a DW of 1 second, $x = 20\%$ and spacing $m = 0.1$ chip. Values used for the length of DW and $x\%$, will be explained in Section 4.2.

Though the RM test detects correlation distortions in both cases, it is easy to observe how the detection events differ for duration and continuity in the two scenarios, presenting a continuous behavior in the case of spoofing (Fig. 4.1) and a more sporadic trend in the presence of multipath (Fig. 4.2). Analyzing Fig. 4.2, we can observe very high variations in the metric M, that may not be not necessarily due to multipath effects, but could also be from other impairments, like signal blockage. In Fig. 4.3 the values of C/N_0 for PRN 3 are presented for To-1 data set, and we can observe that some of the strongest effects are related to big drops of C/N_0. The causes of these drops could be due to any environmental effect.

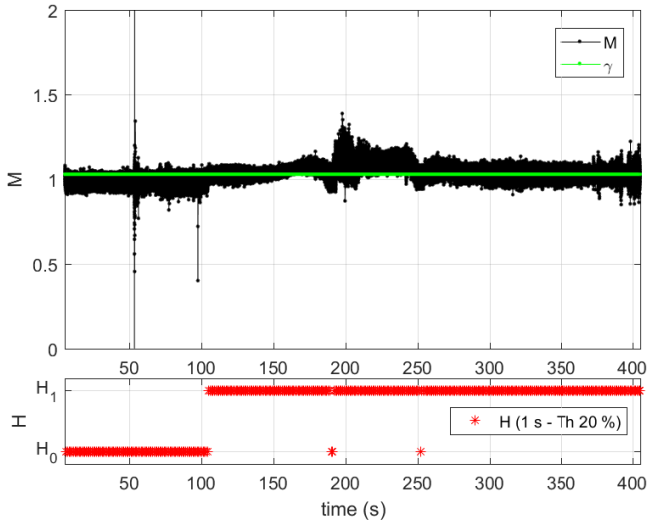

Fig. 4.1 Ratio Metric test: example on a single channel, in the presence of spoofing signal. M and γ vs time (top plot) and correspondent decision vs time (bottom plot). ds6 dataset, PRN 22.

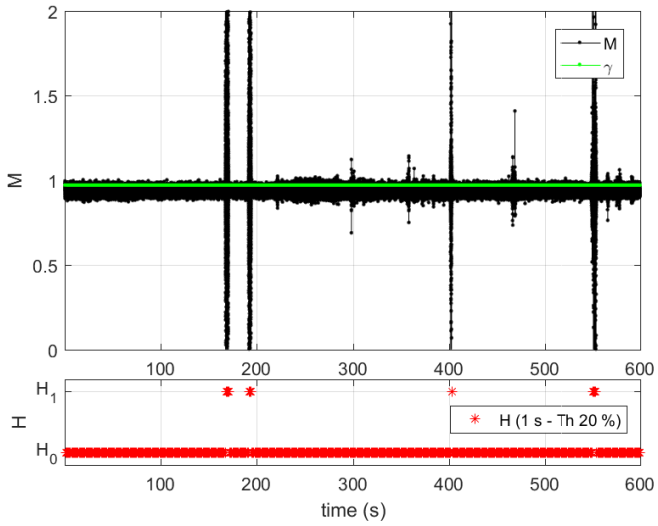

Fig. 4.2 Ratio Metric test: example on a single channel, in the presence of multipath. M and γ vs time (top plot) and correspondent decision vs time (bottom plot). To-1 dataset.

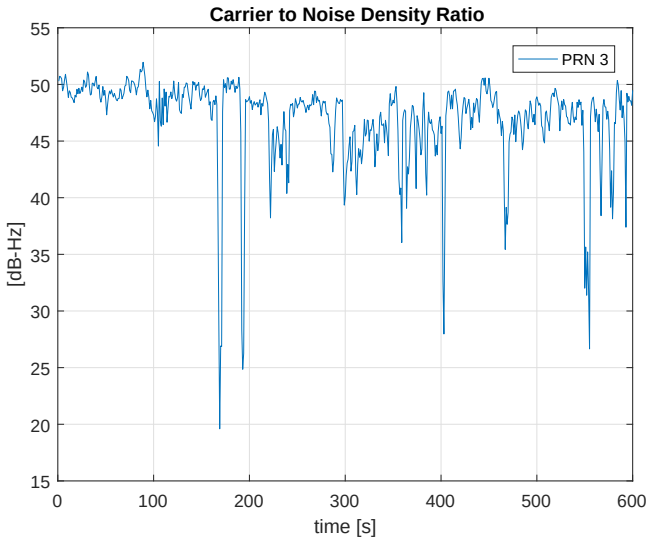

Fig. 4.3 Carrier to Noise density Ratio for satellite signal during dynamic data collection in deep urban scenario. To-1 dataset, PRN 3.

On the other hand, observing Fig. 4.1 we can clearly see a change in the mean value of the metric. This change occurs after 100 seconds into the test, once the spoofer is present and modifies the delay being tracked by the receiver. It is also important to point out that the metric, not only shows distortions in the correlation function, present between 100 and 250 seconds, but it also detects the change of the shape between the satellite-only correlation, present from 0 to 100 seconds, and the spoofing-only correlation, from 250 to 400 seconds. This change, allows for spoofing detection in this specific satellite for the whole duration of the test. In other satellites, the decision may detect distortions only during the phase when the signals are separating, as it can be observed in Fig. 4.4. Behavior shown in Fig. 4.4 is similar to what was observed in Section 3.4, when using a higher x and a wider spacing m.

The RT, as detailed in Section 3.2, is able to take decisions on each single channel, every integration time [67, 81]. Using the algorithm described in Section 3.3, we take decisions on each single channel, every DW. In the next Section, we will see how the difference in decision rate can be exploited to discriminate among impairments.

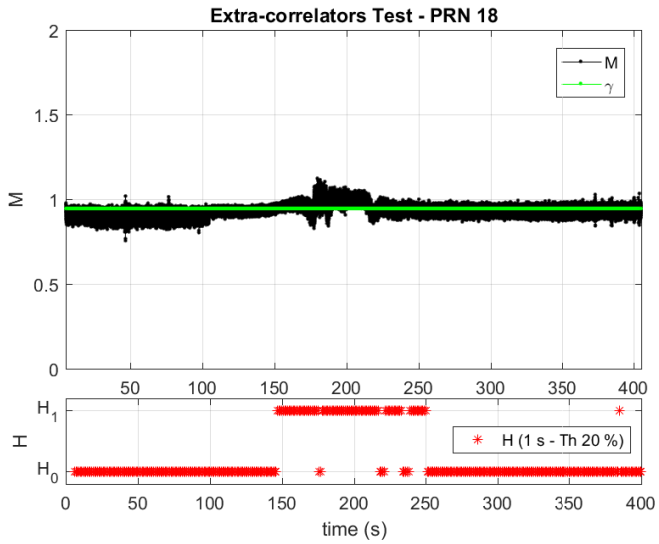

Fig. 4.4 Ratio Metric test: example on a single channel, in the presence of spoofing signal. M and γ vs time (top plot) and correspondent decision vs time (bottom plot). ds6 dataset, PRN 18.

4.2 Building a new metric

What clearly emerges from the previous example is that the RM test can be successfully used for spoofing detection, but it might be characterized by a high number of false alarms given by the presence of other types of impairments that distort the correlation function in a similar way. Before presenting the metric able to discriminate between the two phenomena, some considerations have to be done and some parameters introduced. In general, the effects of an intermediate spoofing attack on the receiver are characterized by a unique combination of continuity over time and number of satellites simultaneously impaired.

Taking these considerations into account, two parameters that will be used in the definition of the new detection metric, can be introduced hereafter:

- l_{DW} is the duration of the DW. This value has to be set by trading off between the variability rate of the events and the decision rate wanted.
- $d_x^i(t_k)$ is the binary output of the decision taken on the k-th DW. It is equal to 1 in the case a disturbance is detected in the current DW and 0 otherwise. For

the i-th satellite, $d_x^i(t_k)$ can be defined as:

$$d_x^i(t_k) = \begin{cases} 1 \text{ if } M \geq \gamma \text{ for } x\% \text{ of } l_{\text{DW}} \\ 0 \text{ otherwise} \end{cases} \tag{4.1}$$

where t_k is the time instant that corresponds to the end of the k-th DW. In other words, in a single DW, the decision on the status of a single satellite signal is taken after fixing the percentage of time in which the RM overcomes the threshold as explained in Section 3.3.

As an example, for the i-th satellite, $d_{20}^i(t_k)$ means that the signal is declared *impaired* in the k-th DW if $M \geq \gamma$ for, at least, the 20% of l_{DW}. Such a parameter is fundamental in the decision process, and in particular, in discriminating among impairments.

The effects of a spoofing attack tend to be continuous over time and to simultaneously affect more than one satellite, while multipath events have typically a much faster variability, especially in dynamic urban scenarios. In order to investigate these aspects on the examples shown in Fig. 4.1 and Fig. 4.2, we introduce the variable

$$N_{\text{sat},x}^I(t_k) = \sum_{i=1}^{N_{\text{sat}}} d_x^i(t_k) \tag{4.2}$$

where N_{sat} is the number of satellites in view. Such a variable is useful since, in real scenarios affected by multipath reflections, given a specific satellite geometry, it will be unlikely that more than two or three satellites have multipath components at the same time instants, unless in case of signal blockage. On the other hand, if the receiver is under a realistic intermediate spoofing attack, the probability of detecting anomalies in all the tracked channels (or at least on a large subset) is high.

We then consider its behavior in the two scenarios ds6 and To-1, for different values of x. Fig. 4.5 and Fig. 4.6 show the trend of $N_{\text{sat},x}^I(t_k)$ for $x = 20\%$ and $x = 50\%$, with a l_{DW} of 1 s and $N_{\text{sat}} = 7$.

In the case of scenario ds6 (Fig. 4.5), α changes depending on the value of x, and we can observe the impact of the larger x. Nevertheless, if we observe the changes for the multipath scenario, in Fig. 4.6, the changes are much more evident, to the extreme where no impairments are declared for $x = 50\%$. In addition, in the former

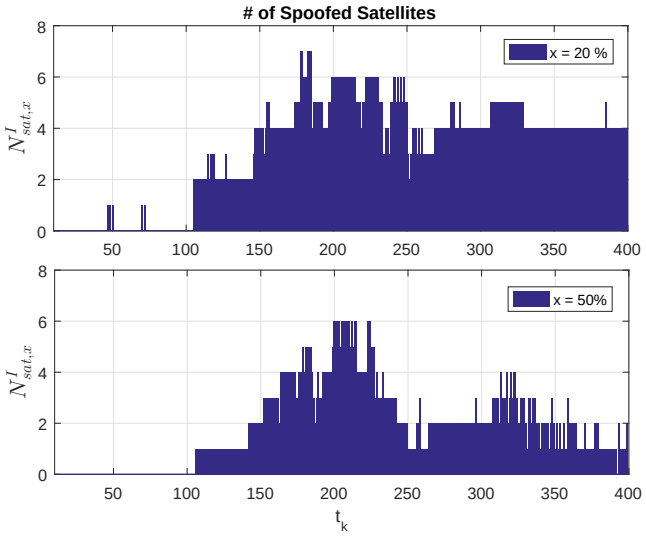

Fig. 4.5 α vs time. Comparison between two values of x: 20% (top plot) and 50% (bottom plot). $l_{\mathrm{DW}} = 1$ s. $N_{\mathrm{sat}} = 7$. Scenario Txb-6.

case, a large number of satellites are *impaired*, while, in Fig. 4.6, $N^I_{\mathrm{sat},x}(t_k) > 1$ only in correspondence to few t_k.

We can now define variable δ as:

$$\delta = \frac{\sum_k d^i_{x_1}(t_k) - d^i_{x_2}(t_k)}{K_{x_1}} \tag{4.3}$$

where $x_2 > x_1$ and K_{x_1} is the number of DWs in which there is at least one satellite impaired, when $x = x_1$, for the duration of the test. The behavior of δ is shown in Fig. 4.7 and Fig. 4.8 for scenarios ds6 and To-1 respectively. In these Figures, $x_1 = 20\%$ and x_2 varies in the interval $[20, 60]$. In the case of spoofing, δ is affected linearly with the variations of x_2 and in the case of urban environment δ has an increasing trend proportional to larger x_2 and then it saturates once no distortions are detected. Looking at these two Figures, we see that $x_2 = 50\%$ is indeed a good value for x_2 in order to increase multipath discrimation. As observed in Fig. 4.7, increasing the value of x_2 would not provide advantages to the multipath discrimination, and would increase the spoofing δ. In other words, with these results we observe that there are two key elements able to highlight the differences between the two data

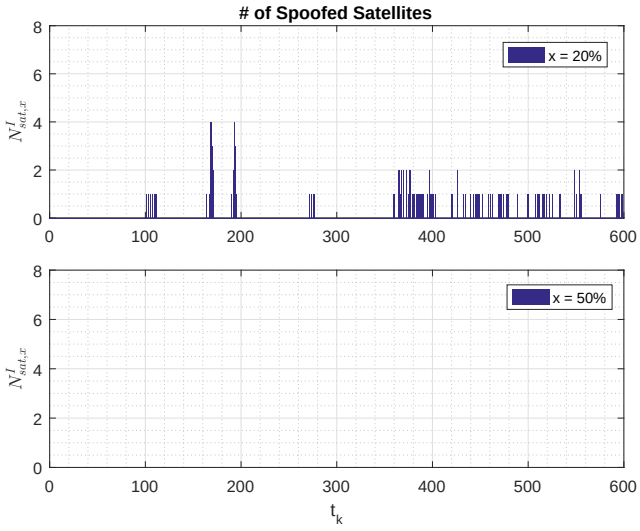

Fig. 4.6 α vs time. Comparison between two values of x: 20% (top plot) and 50% (bottom plot). $l_{\mathrm{DW}} = 1$ s. $N_{\mathrm{sat}} = 7$. Scenario To-1.

collections: the number of satellites contemporaneously impaired and the duration of the impairment over consecutive DWs.

4.3 Definition of Beta

Taking into consideration the parameters previously introduced, we are ready now to define the multidimensional metric for the detection in a single DW. If we introduce

$$s(t_k) = \sum_{i=1}^{N_{\mathrm{sat}}} \frac{d^i_{x_1}(t_k) + d^i_{x_2}(t_k)}{\max(N^{\mathrm{I}}_{\mathrm{sat},x_1}, 1)} \tag{4.4}$$

then the full detection variable is

$$\beta(t_n) = \frac{1}{N_{\mathrm{W}}} \sum_{k=n-N_{\mathrm{W}}+1}^{n} s(t_k) \tag{4.5}$$

where N_{W} is the number of DWs used to observe the continuity of the event. In fact, in order to further reduce the number of false alarms of the detection process $\beta(t_n)$ is obtained as an average of $s(t_k)$ over N_{W} consecutive DWs.

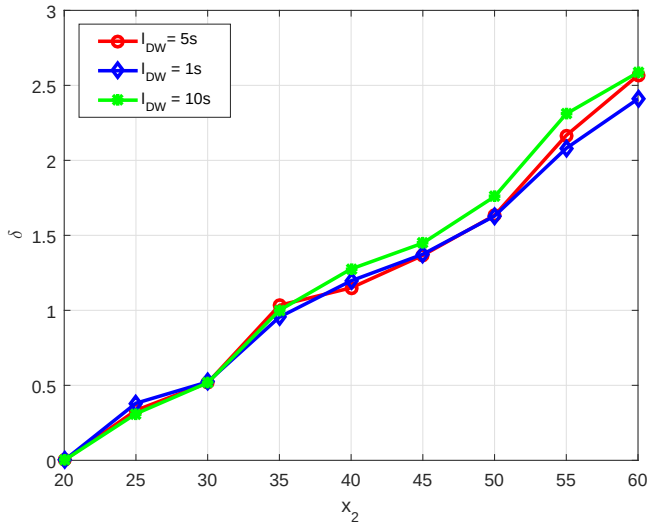

Fig. 4.7 δ vs x_2, for different l_{DW}. ds6 scenario.

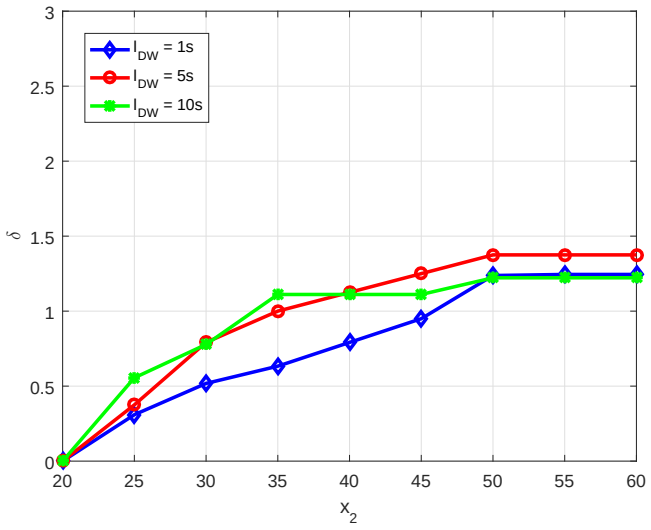

Fig. 4.8 δ vs x_2, for different l_{DW}. To-1 scenario.

β is a discrete variable, evaluated at each decision instant t_n, with $t_n - t_{n-1} = l_{DW} \cdot N_W$. It assumes discrete values that depend on the parameters N_{sat} and N_W, and it is always included in the range $[0,2]$. In fact, $\beta(t_n) = 0$ only if $d^i_{x_1}(t_k) = 0$, $\forall i = 1,..,N_{sat}$, and $\forall k = n - N_W + 1,...,n$ (i.e., for all the tracked satellites and in any observed DW, M does not overcome the threshold γ for more then $x_1\%$ of the time). On the contrary, if $\beta(t_n) \neq 0$, its smallest possible value is $\frac{1}{N_W}$, independently from N_{sat}, while β is equal to 2 (i.e., its biggest possible value) only if $d^i_{x_2}(t_k) = 1$, $\forall i = 1,..,N_{sat}$, and $\forall k = n - N_W + 1,...,n$ (in other words, if, for all the impaired satellites and in any observed DW, M overcomes the threshold γ for at least the $x_2\%$ of the time).

As for an example, Fig. 4.9 shows the sets of possible values for β, in the case of $N_W = 3$ and N_{sat} in the range $[3,8]$.

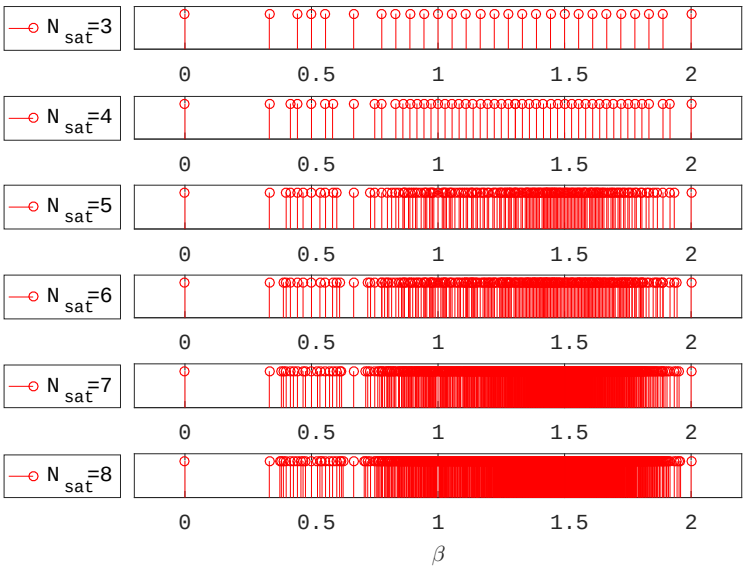

Fig. 4.9 Set of possible values for the β metric, in the case of $N_W = 3$ and N_{sat} in the range $[3,8]$.

In order to use β to take the decision on both the presence and the type of disturbance, we need to fix three thresholds:

- γ to compare M against. As stated in Section 3.2, γ is based on the desired $P_{fa,M}$, and it is obtained using (3.9). As we will see in section 4.3.1, the $P_{fa,M}$,

used to obtain γ, can be directly set on the basis of the false alarm probability of β, $P_{\text{fa},\beta}$.

- γ_1 to decide between H_0 and H_1. As it is clear from the analysis summarized in Fig. 4.9, if, at the time instant t_n, at least one satellite is declared impaired in a DW, β is different from 0 and assumes values equal or greater to $\frac{1}{N_W}$. Therefore, the condition $\beta < \gamma_1 = \frac{1}{N_W}$ is used to decide whether or not the signal is distorted;

- γ_2 to decide between spoofing and multipath. While γ and γ_1 are fixed on the basis of the statistical characteristics of β, an empirical approach will be followed to set the threshold γ_2. To do so, some preliminary considerations are needed.

  Let's first consider a multipath scenario, in which the effects on the satellitez signals appear over consecutive detection windows. In this case, it is likely to have a sub-set of satellites $\left(N^{\text{I}}_{\text{sat},x_1}\right)$ in each detection window that are declared impaired only for $x = x_1$ and β tends to be equal to 1. In the limiting case (i.e., if $d^i_{x_2} = 0\ \forall i = 1,...,N^{\text{I}}_{\text{sat},x_1}$), $\beta = 1$. On the other hand, under a spoofing attack able to control the majority, or all, of the satellites in view, β approaches its maximum. In the limiting case, in fact, when all the satellites in view are spoofed in all the detection windows (i.e., if $d^i_{x_2} = 1\ \forall i = 1,...,N^{\text{I}}_{\text{sat},x_1}\ \forall w = 1,...,N_W$), $\beta = 2$. A balanced value for γ_2 can be fixed by considering (as a limit to decide for multipath and not for spoofing) a heavy multipath scenario (i.e., $d^i_{x_1} = 1\ \forall i = 1,...,N_{\text{sat}}$ in any considered DW), in which only a small portion of the impaired satellites overcomes the threshold also for $x = x_2$. If such a portion is limited to the 25% of the impaired satellites (i.e., $d^i_{x_2} = 1\ \forall i = 1,...,N_{\text{sat},2}$, with $N_{\text{sat},2} \leq \frac{N^{\text{I}}_{\text{sat},x_1}}{4}$ in any DW), β results to $5/4$. Taking all these considerations into account γ_2 is set at $5/4$ in all the experiments presented hereafter.

4.3.1 Statistical Characteristics of β

In order to fix the value of the threshold γ used in the tests, we need to describe the metric in terms of its statistical characteristics. In our case, we need to write the false

alarm probability of β, i.e.,

$$P_{\text{fa},\beta} = P\left(\beta \neq 0 \mid H_0\right) \tag{4.6}$$

as a function of the false alarm probability of M

$$P_{\text{fa},M} = P\left(M \geq \gamma \mid H_0\right) = \int_{\gamma}^{\infty} f_M(m \mid H_0)\,\mathrm{d}m \tag{4.7}$$

that in the case of high C/N_0 assumes the form $P_{\text{fa},M} = \frac{1}{2}\text{erfc}\left(\frac{\gamma-\mu_0}{\sqrt{2}\sigma}\right)$.

To do so, let us proceed step by step following the structure of β and considering that the analysis hereafter is written under the hypothesis H_0. At the time instant t_k:

- d_x^i can assume only the two values $\{0,1\}$ in accordance with Eq. (4.1). The probability of false alarm of d_x^i is the probability that M overcomes the threshold γ for at least $x\%$ of the time in the current DW. In a single DW the decision process on d_x^i can be represented by a Bernoulli process. In fact, indicating with L_T the total number of decisions on M in a single DW, we have a finite sequence of random variable that assume the value 1 with probability $P_{\text{fa},M}$ and the value 0 with probability $(1-P_{\text{fa},M})$. $d_x^i = 1$ if we obtain L_x *successes* (defined by the condition $M \geq \gamma$) in L_T trials, being $L_x = \lfloor \frac{xL_\text{T}}{100} \rfloor$. The false alarm probability of d_x^i can be then written as

$$P_{\text{fa},d_x^i} = P(d_x^i = 1 \mid H_0) = \sum_{L_i=L_x}^{L_\text{T}} \binom{L_\text{T}}{L_i} P_{\text{fa},M}^{L_i} \left[1 - P_{\text{fa},M}\right]^{(L_\text{T}-L_i)} \tag{4.8}$$

- referring to (4.5) and placing $D^i = d_{x_1}^i + d_{x_2}^i$, D^i can assume the values $\{0,1,2\}$, depending on the values of $d_{x_1}^i$ and $d_{x_2}^i$ at the instant t_k. The false alarm probability on D^i becomes

$$\begin{aligned} P_{\text{fa},D^i} &= P\left(D^i \neq 0 \mid H_0\right) = P\left(D^i = 1 \mid H_0\right) + P\left(D^i = 2 \mid H_0\right) = \\ &= P\left(d_{x_1}^i = 1 \wedge d_{x_2}^i = 0 \mid H_0\right) + P\left(d_{x_1}^i = 1 \wedge d_{x_2}^i = 1 \mid H_0\right) = \\ &= P_{\text{fa},d_{x_1}^i} \end{aligned} \tag{4.9}$$

- considering now $s(t_k)$, if:

  - $\forall i, i = 1,\ldots,N_\text{sat}: \quad D^i = 0 \quad \Longrightarrow \quad s = 0;$

- $\exists i,\, i = 1,...,N_{\text{sat}}: \quad D^i \neq 0 \quad \forall i = 1,...,N_{\text{sat}} \quad \Longrightarrow \quad s = \sum_{i=1}^{N_{\text{sat}}} \frac{D^i}{N^{\text{I}}_{\text{sat},x_1}}$ and it can assume the values $\left\{1 + \frac{z}{N^{\text{I}}_{\text{sat},x_1}}\right\}$, with $z = 0,1,...,N^{\text{I}}_{\text{sat},x_1}$.

Analogously to what was done for d^i_x, the false alarm probability $P_{\text{fa},s}$ can be written by modeling s as a Bernoulli process in which a *successful* event (in our case, the detection of an impaired satellite, i.e., $D^i \neq 0$) occurs with probability $P_{\text{fa},d^i_{x_1}}$:

$$P_{\text{fa},s} = \sum_{N_i=1}^{N_{\text{sat}}} \binom{N_{\text{sat}}}{N_i} P^{N_i}_{\text{fa},d^i_{x_1}} \left[1 - P_{\text{fa},d^i_{x_1}}\right]^{(N_{\text{sat}} - N_i)} \tag{4.10}$$

- Finally, at each time instant t_n, if:
  - $\forall k,\, k = n - N_{\text{W}} + 1,...,n: \quad s = 0 \quad \Longrightarrow \quad \beta = 0;$
  - $\exists k,\, k = n - N_{\text{W}} + 1,...,n: \quad s \neq 0 \quad \Longrightarrow \quad \beta(t_n) = \frac{1}{N_{\text{W}}} \sum_{k=n-N_{\text{W}}+1}^{n} s(t_k)$ and it assumes values between 0 and 2.

Modeling again the decision in each DW as a Bernoulli event with probability of success (i.e., $s \neq 0$) of $P_{\text{fa},s}$, the false alarm probability of β becomes

$$P_{\text{fa},\beta} = \sum_{w=1}^{N_{\text{W}}} P^w_{\text{fa},s} \left[1 - P_{\text{fa},s}\right]^{(N_{\text{W}} - w)} \tag{4.11}$$

As an example, the behavior of β is plotted in Fig. 4.10 as a function of $P_{\text{fa},M}$ and x_2, for $N_{\text{sat}} = 10$, $N_{\text{W}} = 5$ and $x_1 = 10\%$.

After fixing the desired probability of false alarm of β, it is possible to obtain the maximum probability of false alarm of M, and then, using (3.9), fix the threshold γ. Following the previous example, Fig. 4.11 indicates the maximum values of $P_{\text{fa},M}$, by varying x_2 in the range $[10, 50]$ and after fixing $P_{\text{fa},\beta}$ at 10^{-5}.

Observing Figs. 4.11 and 4.10, we understand that we can set a high probability of false alarm on $P_{\text{fa},M}$ and still obtain a very low $P_{\text{fa},\beta}$. With this we conclude the statistical analysis of metric β and we have the basis to set the parameters accordingly.

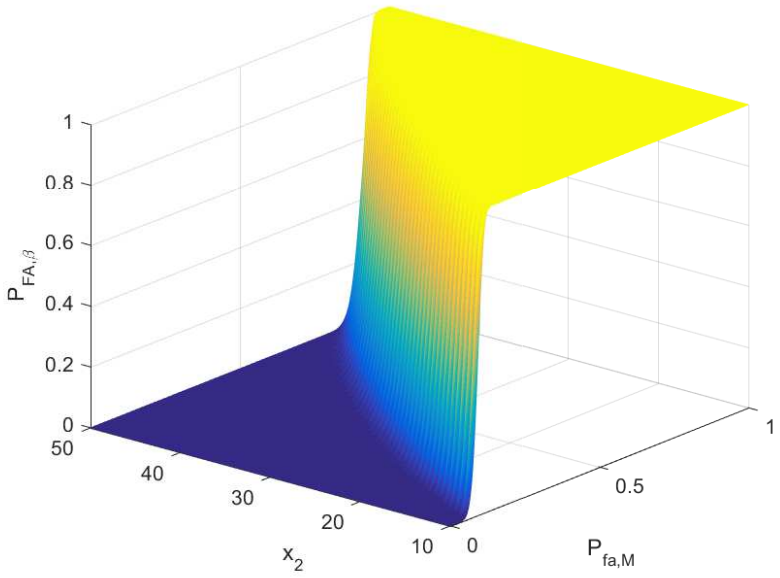

Fig. 4.10 Behavior of $P_{\text{fa},\beta}$ as a function of $P_{\text{fa},M}$ and x_2. $N_{\text{sat}} = 10$, $N_{\text{W}} = 5$ and $x_1 = 10\%$

4.4 Testing the new metric

The capabilities of β have been tested against different scenarios, using the parameters summarized in Table 4.2.

Table 4.2 β Parameters

Parameter	Value
x_1	20%
x_2	50%
l_{DW}	1 s
N_{W}	5
E-L spacing	0.2 chip
$P_{\text{fa},M}$	10^{-2}
$P_{\text{fa},\beta}$	$< 10^{-20}$

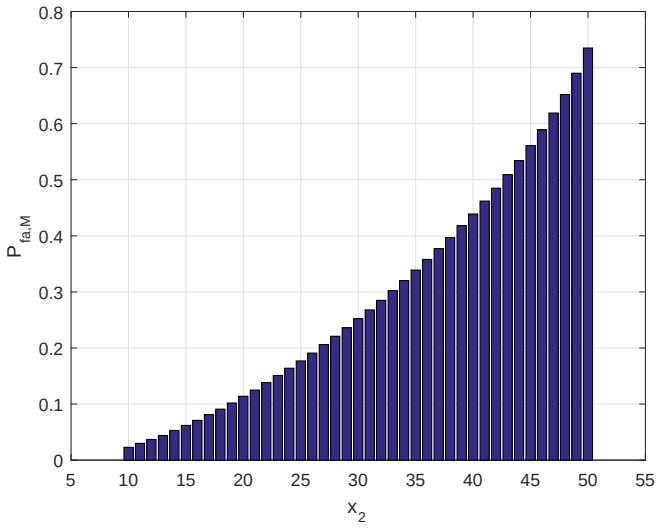

Fig. 4.11 Maximum values of $P_{\text{fa},M}$ vs x_2. $P_{\text{fa},\beta} = 10^{-5}$. $N_{\text{sat}} = 10$. $N_{\text{W}} = 5$.

Results for ds6 and To-1

The results of β for the data collections ds6 and To-1 are presented in Figs. 4.12 and 4.13 respectively. From the β trend it can be observed that the decision is taken correctly, according to the scenario: spoofer for ds6 and multipath for To-1.

In Fig. 4.12 we observe that the spoofing attack goes from 110 s until the end of the test, during this time the spoofer takes control of the receiver and drives it away as shown in [38]. It can also be observed that during the first part of the dataset, a small multipath is detected which can be justified due to the dynamic nature of the test.

From the results in Fig. 4.13, we observe a variable behavior on β, that takes values between 0 and 1 during the whole duration of the test. This behavior is in line to what was expected for heavy multipath scenarios.

In order to further validate the method, hereafter we present the results of β over a selection of different scenarios, mentioned in Table 4.3 and described in Appendix A.

The TEXBAT datasets ds3, ds7 and Clean Dynamic are introduced in Appendix A.1 and are thoroughly described in [38, 35]. To-02 is another dynamic data col-

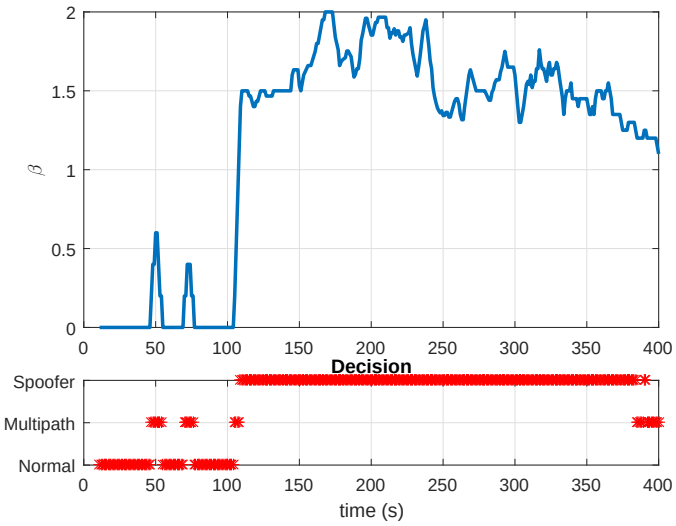

Fig. 4.12 β behavior for ds6 and associated decision

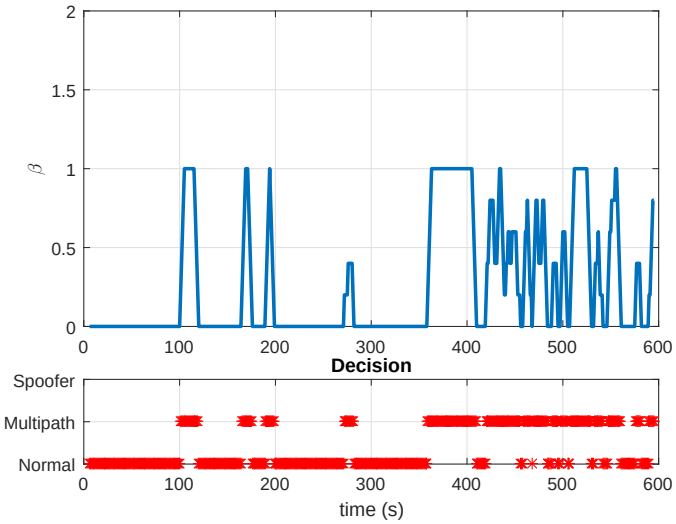

Fig. 4.13 β behavior for To-1 and asociated decision

Table 4.3 Description of the scenarios used for validation of metric β

Name	Place	Date	Description
ds3	Austin, Texas	01/2011	Static matched-power time push
ds7	Austin, Texas	01/2011	Static matched-power evolved time push
To-2	Turin, Italy	02/2015	Dynamic urban
Clean Dynamic	Austin, Texas	01/2011	Clean Dynamic

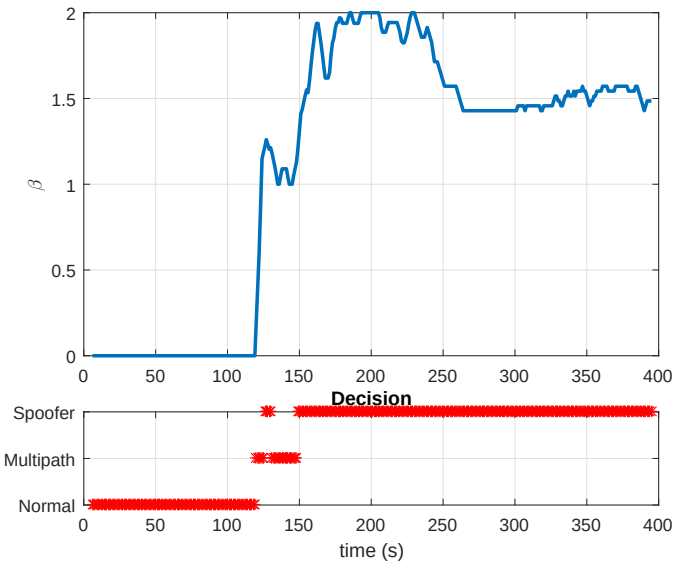

Fig. 4.14 β behavior for ds3 and associated decision

lection, collected in downtown Turin, in which events of multipath are likely and is detailed in Appendix A.3.

Results for ds3

The trend of β for the ds3 is shown in Fig. 4.14. Also in this case, β is able to detect the spoofing from second 115 to the end of the dataset. This result demonstrate the capabilities of spoofing detection in static scenarios. We observe that during the first 30 seconds of the test, the metric reveals the spoofer presence and then falls back to multipath declaration. This is due to the small separation that the spoofer and satellite signal have, making it difficult to distinguish asymmetries. Nevertheless, once we have one *spoofer* flag ($\beta > 1.25$), the receiver should be warned heavily and the navigation solution should not be trusted, even if β falls back to a value of 1. For this scenario, the PVT solution is affected mainly in the time domain, where errors of up to 600 meters are introduced. Additionally, the 3D solution is also affected with errors of up to 300 meters that are caused because not all the satellites are successfully controlled by the spoofing attack at the same time.

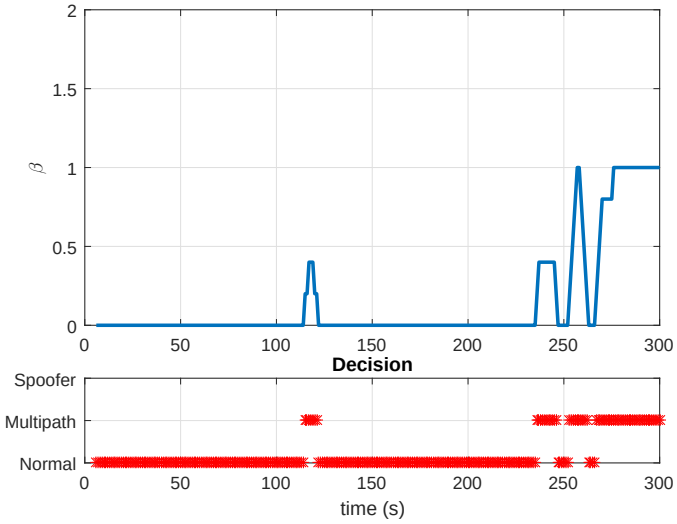

Fig. 4.15 β behavior for To-2 and associated decision

Results for To-2

In Fig. 4.15 the results of β on the data collection To-2 are presented. At the beginning of the test not many impairments are present, but after 200 s, heavy multipath is observed. However, β is never close to 2 as in the spoofer cases, allowing for the correct discrimination between the two types of disturbances. Due to the lack of a reference track is difficult to identify how much error is introduced in the PVT solution by the multipath effects. Nevertheless, observing the navigation solutions for both To-1 and To-2, we see differences of hundreds of meters between consecutive 3D rms positions when the multipath is present. The differences in the real positions are never greater than a tenths of meters.

Results for Clean Dynamic

In Fig. 4.16 we present the results for β against the Clean Dynamic scenario. We can observe some small distortions are present throughout the file, that could be originated from a dynamic urban environment. Also, we can observe the effect at around 50 seconds, which is also present in the case of ds6 scenario (see Fig. 4.12).

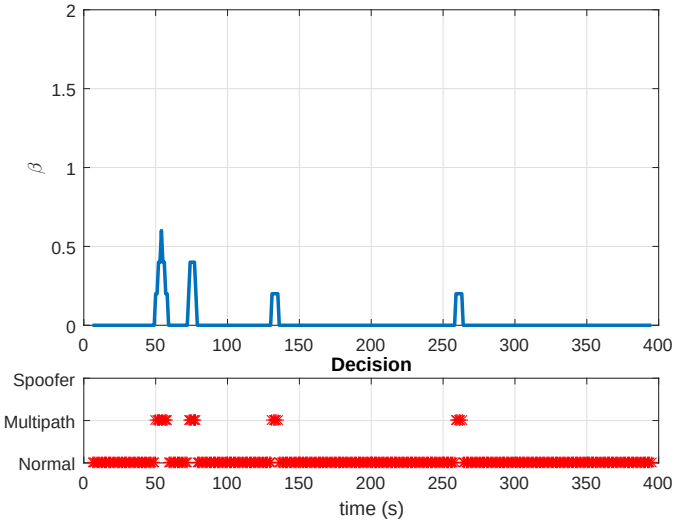

Fig. 4.16 β behavior for Clean Dynamic scenario and associated decision

Results for ds7

In order to further demonstrate that β is effective as a spoofing detection technique, results from the processing of Scenario ds7 of the TEXBAT are also presented. As observed in Fig. 4.17, β detects the spoofer presence starting around 175 s, once the push-off phase of the spoofer has started. At the beginning of the attack (120-160 s) distortions are not detected, due to the alignment between the signals. In this case the technique would be vulnerable to the navigation data bit changes if no other checks are done. Nevertheless, once the spoofer pushes away the real signal in order to modify the delay, β has a solid value close to 2 for the duration of the attack. In this case, all the satellites are correctly controlled by the spoofing attack and errors of 375 meters (or 1.25 μs) are introduced in the time domain of the PVT solution at the end of test. For this spoofing attack, the 3D positioning error is never greater than 30 meters, thus confirming the correct behavior of the time push.

These results show the detection capabilities of β as an effective spoofing detection technique, given that even with a small distortion of the correlation function, it is able to raise an alarm for the presence of the spoofer.

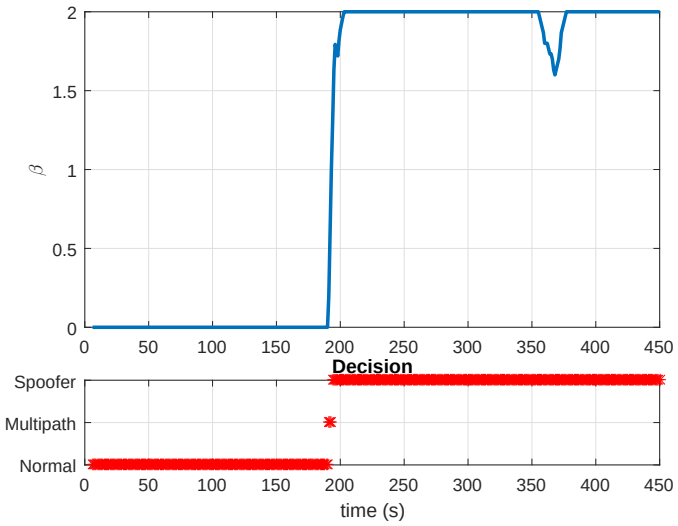

Fig. 4.17 β behavior for dataset ds7

4.5 Effects of correlator spacing

One important parameter for the RM computation is the spacing between early and late correlators, δ_m. Even though one could think that having a large spacing (1 chip), giving a broader window of observation of the distortion, would provide better results, in this Section the advantages and disadvantages of having a narrower spacing (0.2 chip) are discussed.

When considering spoofing attacks, we could say that the noise level of the correlation function during the "no-spoofing" calibration phase will be different from when the spoofing signal is dominant. The spoofing signal will have a higher power level than the satellite signal, thus decreasing the variance of each single correlation and, in consequence, decreasing σ of M. Lowering the noise level, could create a slight shift in the mean value of M, μ_1, as observed in Fig. 4.1. These differences are detected by the ratio test.

Taking a look at Fig. 4.18 we observe how the spacing affects the window of detection of the spoofing attack for scenario ds3. We know that the attack is present after 110 seconds, and last for the entire duration of the dataset. After 250 seconds, the separation between the two correlation functions is wide enough that the correlation function at zero delay is not distorted anymore.

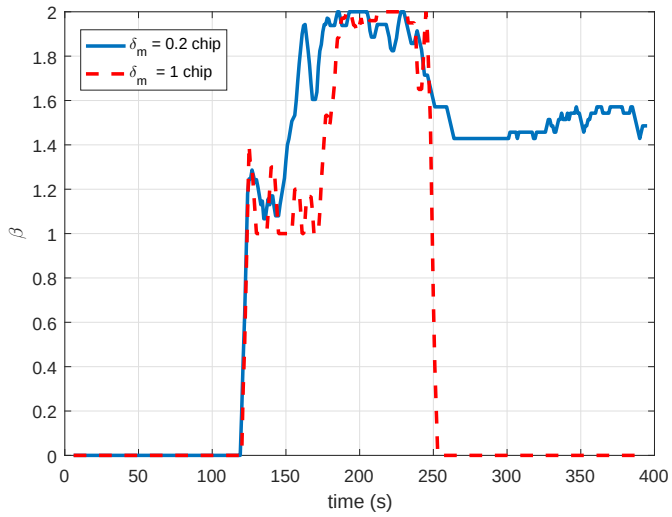

Fig. 4.18 β behavior for scenario ds3 and two different correlation spacing

Nevertheless, if the SQM is observing the correlation function of the spoofing signal, it will have different statistics than the ones obtained during the calibration phase, thus making it prone to detection through the RM. This means that the smaller spacing could improve the distinction between the real signal and the spoofing one, even if it the observed correlation function is not distorted. This is not necessarily always the case as was observed in Fig. 4.4 and is the reason why the value of β, computed with correlation spacing of 0.2 chips, also decreases a little after 250 seconds. Depending on the application, we may be inclined to use a wider correlation or multiple correlator pairs.

Having a narrower correlator spacing can affect the capabilities of detecting multipath signals that are far from the correlation peak, a trade-off between better multipath detection and better spoofing attack detection capabilities needs to be considered when deciding the spacing. A correlator pair closer to the peak is able to mitigate the effects of multipath components [22]. In Fig. 4.19 we observe that the behavior of β in the To-1 scenario, for both spacings, is comparable. For the results shown in this Chapter the Early-Late correlation spacing was set equal to 0.2 chips.

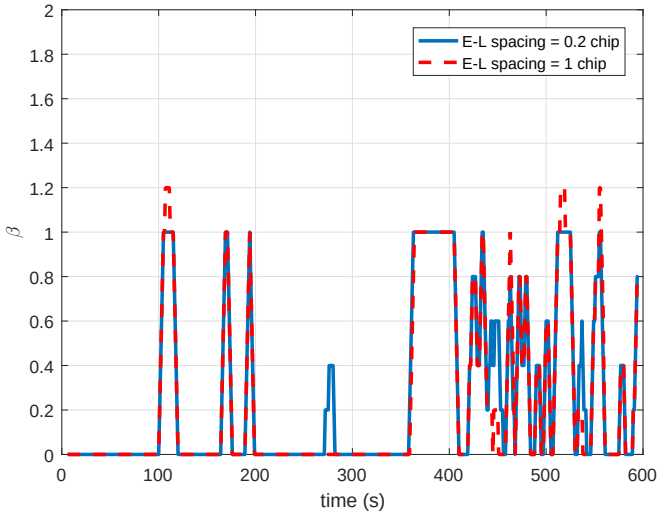

Fig. 4.19 β behavior for To-1 for two different correlation spacing

4.6 Conclusions on the multidimensional ratio metric

From the results presented in Section 4.4, we can observe that only matched-power scenarios were used. It has to be considered that in case of over-powered scenarios, the distortions created in the correlation function are small and not easily detectable. For this reason, in order to maximize the detection probability, it is important to have an additional control on the signal power level, or AGC gain, on top of the SQM. This power measurement control is the easiest way to detect the over-powered type scenarios [84, 3]. A combined approach, using both SQM and AGC is shown in Chapter 5, in which the over-powered cases are detected.

It is also important to clarify that once the metric flags the situation as spoofer presence, the receiver should not trust the navigation solution, until it can confirm that it is not being spoofed anymore. This situation makes the SQM a transient indicator, which means that, it is not capable of detecting the spoofer presence in all the situations, only when the signal is close the satellite signal. The spoofing signal may be present outside the range of observation of the correlators, making it harmless to a locked receiver, but could become harmful in cases of re-acquisition or loss-of-lock of the DLL. This could be fixed by adding an extra scanning correlator, that changes its delay continuously, and flags the presence of unexpected signals throughout the whole correlation space [67, 81].

In this Chapter we have detailed the functional principle of an SQM algorithm and the improvements and external checks that can be added on top of it. With the additional checks, the window where the spoofer presence is not detected is reduced, improving the detection capabilities. The new metric is able to distinguish between multipath events and spoofing attacks, which have always been a matter of concern when using SQM for spoofing detection.

Chapter 5

Detection of overpowered spoofing attacks

In this Chapter, we explore the observation of the AGC gain as spoofing detection strategy for aiding with the detection of overpowered spoofing attacks. The AGC is used in every front-end with the goal of maintaining the distribution of the samples of the incoming signal, thus avoiding saturation or signal loss due to changes on the signal power level.

The motivations of this work are two-fold: first, to show how the AGC measurements can be used to flag overpowered spoofing attacks, that may not be detected by means of the SQM and second, to show how the outputs of a COTS receiver can be used in order to build an effective spoofing detection module.

Additionally, we present a method to discard false alarms on the AGC due to the presence of jammer transmissions that could be wrongly flagged as overpowered spoofing events. The method compares the values of the AGC and C/N_0 variations, that are strongly related when a jamming event is occurring, and shows that that relationship is different in the spoofing context.[1]

[1] The work presented in this Chapter was developed during an exchange period with the University of Colorado at Boulder, working with professor Dennis M. Akos. Datasets were kindly provided by the GPS laboratory at Stanford University and Zeta Associates.

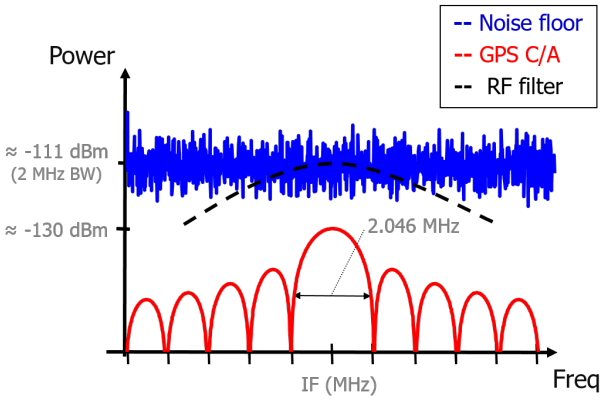

Fig. 5.1 Power level comparison between the noise floor of the receiver, the effect of the RF front-end filter and the GPS C/A signal. *Courtesy of Dr. Dennis Akos*

5.1 Power measurement monitoring

Power measurement monitoring was originally presented in [7] as an interference monitoring system and then in [3] as a spoofing detection method. It is based on the observation of the response of the AGC, in order to detect additional signals in the receiver that are not supposed to be present.

Normally, the received power of the GNSS signal is below the thermal noise floor, as can be seen in Fig. 5.1. The incoming power P_N can be written as:

$$P_N = kT_A BW \tag{5.1}$$

where k is the Boltzman's constant, T_A is the effective temperature of the antenna, and BW is the bandwidth of the signal. If we combine the received power, with the first stage of the front end, we obtain:

$$P_N = k(T_A + T_R)BW \tag{5.2}$$

where T_R is the receiver noise temperature and BW is now the bandwidth of the front-end filter.

Fig. 5.2 shows a typical receiver architecture, highlighting the AGC component. The AGC is used to optimize the gain of the front-end, that is the analog part of Fig. 5.2, to the input range of the analog-to-digital converter. The main uses of the

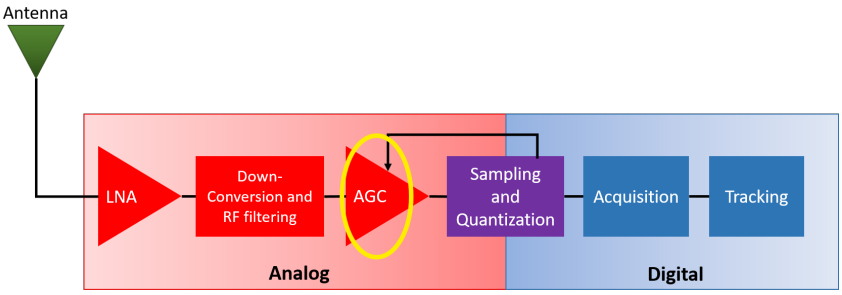

Fig. 5.2 Typical GNSS receiver architecture, highlighting the AGC component, used throughout this Chapter

AGC are for adjusting the gain to the different antennas and to adjust the gain in case of interference that affects the effective power of the GNSS signal. Assuming that the active antenna gain of the receiver is stable, for a given configuration, the AGC will mainly respond to RFI, present in the front-end bandwidth. Thus, it can be effectively used as a monitoring metric.

As discussed in [3], the AGC gain values could present small changes based on the effective temperature of the antenna and on the environment. Nevertheless, high variations in the AGC gain would hint that an additional signal is present in the band.

The AGC gain is a suitable metric for spoofing detection in commercial receivers because many of the COTS receivers available, give access to its value. The AGC is separated from the digital part of the receiver, as can be seen in Fig. 5.2, and this makes it an easy value to have access to, without having to intervene in the signal processing part. In order to asses the capabilities of the AGC gain in a commercial receiver we will use the Novatel G-III receiver and a set of data collections from WAAS stations, described in Appendix A.2.

In a Novatel receiver, the AGC gain measure is called Pulse Width and it has no dimensions. For the following experiments we injected controlled amounts of noise into the receiver in order to be able to translate the values of Pulse Width to quantities equivalent to gain in dB. Fig. 5.3 shows the relationship between the Pulse Width values and the amount of power injected into the receiver. Using the slope of the curves we can translate variations of Pulse Width into variations in terms of dB,

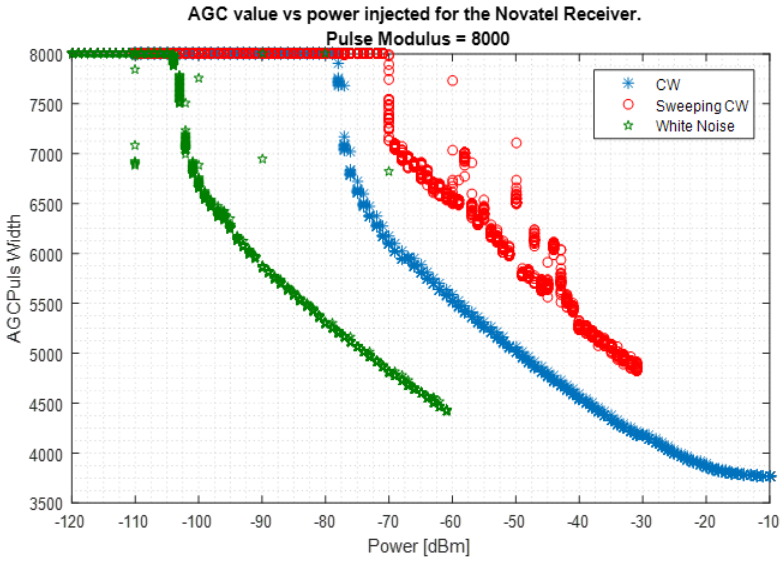

Fig. 5.3 Pulse Width behavior versus injected power for different types of interference. From this figure we can translate the Pulse Width variations into dBs.

that will be used hereafter. The AGC gain is then defined as:

$$G_{AGC} = \frac{\text{Pulse Width}}{\eta} \tag{5.3}$$

where Pulse Width, refers to the output values from the AGC metric of the Novatel receiver, that goes from 8000 in the absence of signal and 3500 indicating the minimum gain, when the AGC reach saturation. Variable η represents the slope of the linear region of the curve.

The AGC gain, referred hereafter as G_{AGC}, is going to be used as a metric for the behavior of the AGC, and thus, for detecting interference events within the receiver's bandwidth. It can be consider a static indicator to assess spoofing, given that no matter if the spoofing signal is aligned or not with the signal tracked by the receiver, G_{AGC} will be able to detect its presence. When G_{AGC} decreases, it means that additional signal power is present in the band, so an interference event is going on.

In Fig. 5.4 we can observe the nominal behavior of the gain, obtained from the Novatel G-III receiver, over 120 hours of data from the Honolulu (HNL) WAAS station.

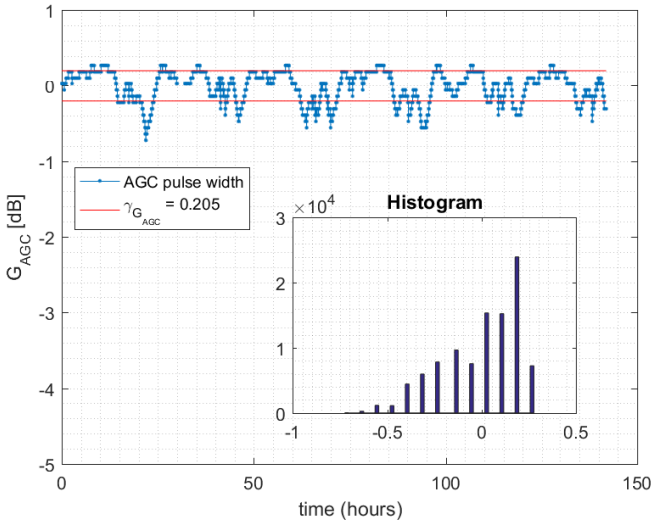

Fig. 5.4 AGC gain example for nominal behavior of the Novatel G-III receiver. Obtained from WAAS station HNL, in Honolulu, Hawaii.

Observing the variance of G_{AGC}, over four different WAAS stations, during periods considered free of interference for more than 480 hours, we obtained that the standard deviation $\sigma_{G_{AGC}} = 0.25$. Using this information we can decide an heuristic threshold and declare that an interference event is occurring when:

$$G_{AGC} < \gamma_{G_{AGC}} \tag{5.4}$$

where $\gamma_{G_{AGC}}$ will be the lower threshold. In our case, $\gamma_{G_{AGC}}$ was conservatively defined as $\gamma_{G_{AGC}} = 4 * \sigma_{G_{AGC}}$, and we obtained a $P_{\text{FA},G_{AGC}} = 2.5e^{-6}$ using this threshold. Metric G_{AGC} can then be effectively used for detecting spoofing attacks as will be shown in Section 5.3.

It is important to notice that during nominal behavior, G_{AGC} have some variations as observed in the Figure. These variations are likely caused by temperature changes in the low noise amplifier (LNA) of the antenna, as they change the efficiency of the internal components of the antenna and front-end. These variations are usually low on WAAS stations, meaning that WAAS stations work well for assessing nominal behaviors in clean and stable scenarios. In other applications, like aviation or dynamic road scenarios, the thresholds may need to be enlarged in order to maintain the probability of false alarm.

Table 5.1 Novatel G-III correlation spacing and Linear combination used

Spacing	-0.1016	-0.0766	-0.0516	-0.025	0	0.025	0.0516	0.0766	0.1016
Lin. Comb.	-1	-1	-1	-1	0	1	1	1	1

The GPS L1 band is classified as Aeronautical Radionavigation Service (ARNS), so it is forbidden for unauthorized systems to transmit additional signals in the frequency band. Nevertheless, illegal transitions inside the band have been detected in several locations. So, it is an important task for enforcing agencies to keep the band clear of interference that could impair critical applications. This aspect will be the focus of Section 5.4.

5.2 SQM using delta test

The SQM technique has been proposed in many different configurations for detection of distortions in the GPS correlation peak, and was explored in depth in Chapters 3 and 4. In this Chapter, we will use a simple SQM metric, that will be called Δ. It will be the difference of the linear combination of multiple correlator values. This metric is simple and easy to use and thus is suitable for implementation in commercial receivers.

The receiver used for these experiments includes multicorrelators pairs at fixed spacings, given by Table 5.1. The table presents the linear combination used. It is important to notice that the metric will be constructed by using observations of the correlation peak from -0.1016 chips to $+0.1016$ chips. The metric is then normalized by the value at zero delay and a *delta metric* is contructed. The *delta metric* was mentioned in Section 3.1.1, and we know that a mean value close to zero is obtained for the nominal correlation shape.

In this case, Δ will be defined as:

$$\Delta = \frac{\mathbf{L}_x - \mathbf{E}_x}{P} \tag{5.5}$$

where $\mathbf{L}_x = (L_{0.10} + L_{0.07} + L_{0.05} + L_{0.02})$ and $\mathbf{E}_x = (E_{0.10} + E_{0.07} + E_{0.05} + E_{0.02})$, representing the linear combination of the late and early correlators, and P is the prompt correlator at zero delay. The behavior of Δ for the WAAS station site HNL over 24 hours can be observed in Fig. 5.5, along with a histogram of the data.

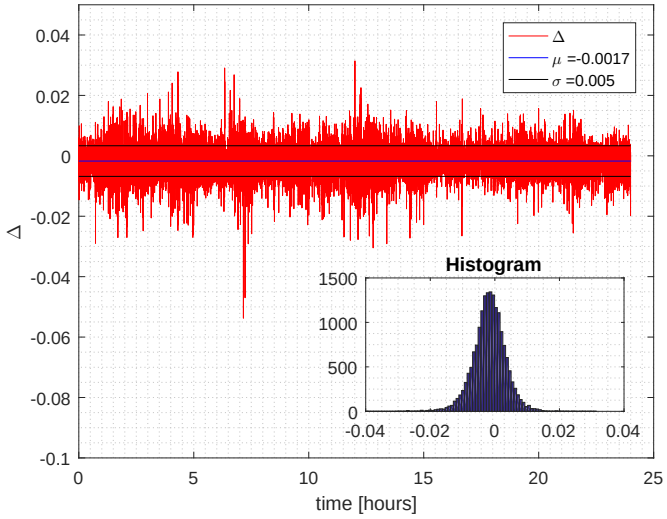

Fig. 5.5 Δ behavior for WAAS station HNL, in Honolulu, Hawaii, over 24 hours. The mean and standard deviation are also highlighted, along with a histogram of the distribution

As explained in Chapter 3, the correlator outputs are distributed as random Gaussian variables with mean value depending on the location of the correlator and its variance depending on the noise components and integration time. The metric Δ is a ratio of Gaussian random variables, thus it is not Gaussian.

The SQM metric is highly dependent on the characteristic of the receiver architecture and on the C/N_0 distribution of the satellites in view. Taking this into account, we can take advantage of the fact that when using a commercial receiver, its internal characteristics will not change between usages or locations, this means that if we can correctly characterize the behavior of the SQM under different C/N_0 distributions and scenarios, the SQM will present the same characteristics independently of the location. In brief, for commercial receivers, we should characterize the behavior under nominal conditions and select thresholds according to the obtained distributions. Afterwards, the thresholds remain fixed and spoofing detection is performed by comparing against them.

With this in mind we provide an example on how to construct an SQM test for spoofing detection, using a COTS receiver's measurements. It is important to notice that in this case, we cannot modify the receiver's structure, nor its parameters. Nevertheless, the type of technique presented here, uses a simple approach that can be easily added on top of any COTS, with the capabilities of providing the required

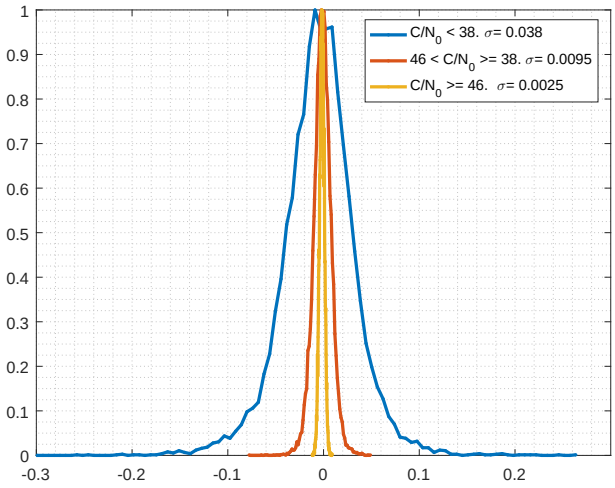

Fig. 5.6 Probability distribution for Δ with different bins of C/N_0, in Honolulu WAAS station. In blue the distribution for satellites with $C/N_0 < 38$dBHz is shown, in red the distribution obtained for satellites with C/N_0 between 38 and 46 dBHz is plotted and in yellow for satellites with $C/N_0 > 46dB-Hz$

measurements. Using COTS receivers, we omit the calibration phase explained in Section 3.3. We instead use the data detailed in Appendix A.2, and due to the high amount of data available, we are able to correctly estimate the distribution for the metric Δ under the hypothesis of no additional signal present. This estimation removes the approximations of Gaussianity of Δ, that was done previously for high values of C/N_0.

As in the other cases, every satellite will provide a metric Δ with a slightly different distribution statistics, which will be dependent on the C/N_0 value, thus requiring computation of thresholds for each of them individually. A different approach would be to separate the satellites inside different C/N_0 ranges and obtain the corresponding distributions and threshold for each separate bin. In Fig. 5.6, the effects of the C/N_0 on the variance of the distribution can be observed. In the Figure, we can see how the variance is greatly reduced with the increase of C/N_0 values.

For the results shown hereafter, we decided to take a conservative approach, using the mean value of all the available metrics Δ, thus considering both high and

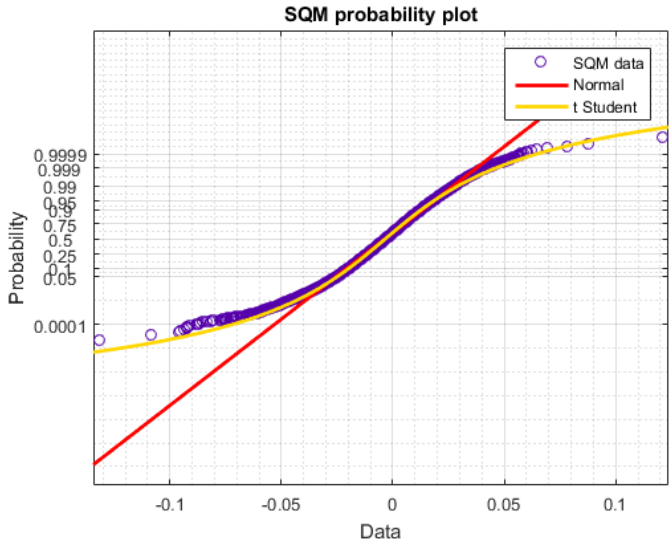

Fig. 5.7 Probability plot for Δ with different fits, the normal distribution (red) and Student's t distribution (yellow)

low C/N_0 values. Our value of Δ will be defined as:

$$\Delta(t) = \sum_{i=1}^{Nsat} \Delta_i(t) \tag{5.6}$$

where $Nsat$ is the total number of satellites available and Δ_i is the metric obtained from satellite i. In this way, we take a decision on the basis of all the signals present in the receiver and computing the mean value. Different approaches, such as the one proposed in Chapter 4, could be adopted instead.

From the observations of the WAAS data and making a distribution fit in Matlab, we obtained that Δ is distributed as a Student's t location scale distribution with nine degrees of freedom and standard deviation $\sigma = 0.005$ as can be observed by Fig. 5.7.

Knowing the distribution of Δ, we can define a two hypothesis test similar to the one proposed in Chapter 3. In this case, we assume that in the absence of interference, the *null hypothesis* H_0 will be:

$$H_0 : \Delta \sim t(\mu, \upsilon) \tag{5.7}$$

where t indicates the Student's t distribution with mean value μ and υ degrees of freedoms obtained from the WAAS data. Following this, we can assume that anything that is not likely to belong to this distribution will indicate the presence of spoofing signals. Extrapolating from this distribution and using the desired probability of false alarm, we can calculate the threshold γ_Δ and the alternate hypothesis will be defined as:

$$H_1 : \text{ not } \Delta \sim t(\mu, \upsilon) \tag{5.8}$$

and the Decision, will be taken as

$$\Delta \begin{cases} \mu - \gamma_\Delta < \Delta < \mu + \gamma_\Delta \longrightarrow H_0 \\ \Delta < \mu - \gamma_\Delta \text{ or } \Delta > \mu + \gamma_\Delta \longrightarrow H_1 \end{cases} \tag{5.9}$$

For the given distribution, and using a two-tail probability table, we obtain the threshold based on the Pfa:

$$\text{for } P_{\text{FA}} = 10^{-5} \longrightarrow \gamma_\Delta = 0.0416 \tag{5.10}$$

Figure 5.8 shows the distribution of the samples with the respective thresholds for P_{FA} of 10^{-5}. Using these thresholds, in Section 5.3, we asses the capabilities of the two metrics Δ and PW to detect spoofing attacks by means of the TEXBAT datasets.

5.3 Baseline results

After having computed the required threshold for the Novatel G-III using the WAAS station's datasets mentioned in Table A.3, we asses the spoofing detection performance of the two metrics, G_{AGC} and Δ, by means of the TEXBAT datasets.

The TEXBAT [38] is discussed in Appendix A.1 and have been used throughout this thesis. For this application we will focus our analysis using the static datasets, given that the baseline thresholds were obtained for static locations, but a similar procedure can be performed for the dynamic tests. The raw data was re-transmitted into the Novatel G-III receiver, using cable connection and the configuration shown in Fig. 5.9. For the initial assessment, the noise source was put to zero, so no

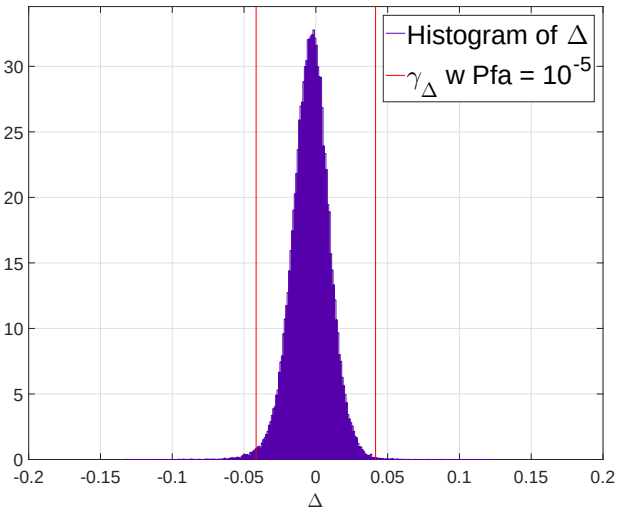

Fig. 5.8 Histogram of Δ with threshold plotted for a P_{FA} of 10^{-5}

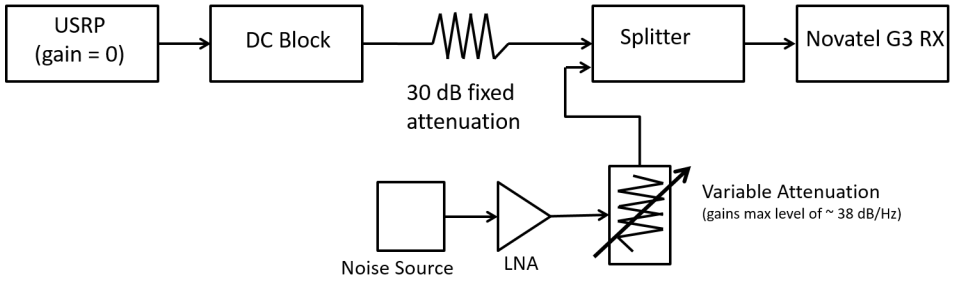

Fig. 5.9 Scheme of the replay of Texbat datasets into the Novatel G-III receiver

additional noise was added. The datasets re-transmitted were all the static datasets, except ds1, which are detailed in Section A.1.

5.3.1 TEXBAT processing

Using the replayed datasets of the TEXBAT, we extracted metrics Δ and G_{AGC} for each scenario and present the results hereafter.

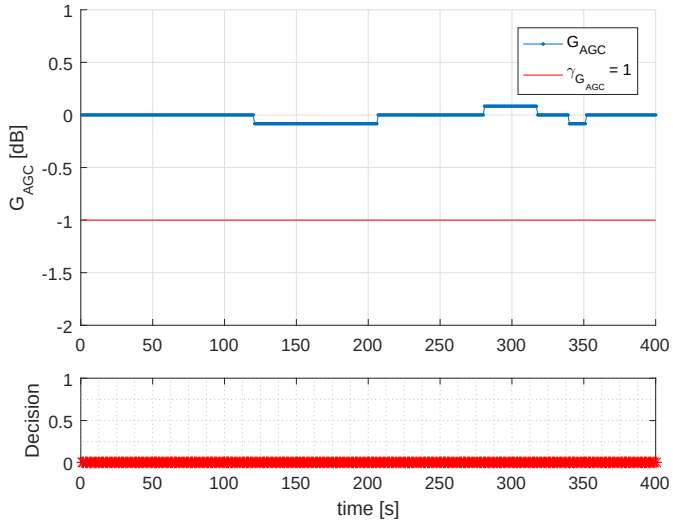

Fig. 5.10 Trend for G_{AGC}, along with the decision taken for the Clean Static dataset. As expected no false alarms are present

Clean Static scenario

In Figs. 5.10 and 5.11, we can observe the results for G_{AGC} and Δ, respectively, for the Clean Static. As expected, no false alarms are triggered and the metrics are well between their thresholds.

It is important to remember the middle ground approach taken for the threshold calculation of the SQM. Given that the TEXBAT datasets include mainly satellites with very high C/N_0, the threshold for the SQM will appear exaggerated and detection of the spoofer could be improved by tightening them.

Scenario ds2

In Figs. 5.12 and 5.13, we can observe the results for G_{AGC} and Δ, respectively, for the spoofed dataset ds2. During the creation of the overpowered TEXBAT datasets, extra noise was added to the signals in order to maintain the C/N_0 level and bury the satellite signal under the noise [38]. Given this, the satellite signal is buried under the noise level and does not create significant distortions in the correlation peak. Thus, the SQM metric Δ detects the presence of asymmetries in the correlation function only a handful of times and a false alarm is detected around 1.2 minutes. We

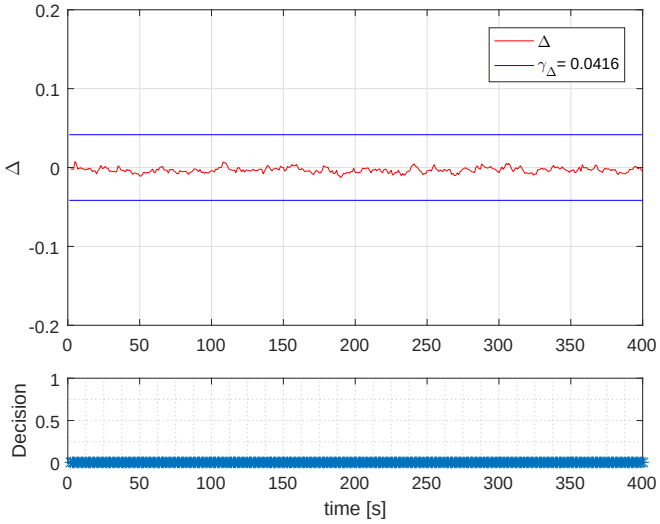

Fig. 5.11 Trend for Δ, along with the decision taken for the Clean Static dataset. As expected no false alarms are present

hypothesize the false alarms triggered in the various TEXBAT data sets are an artifact of the creation of the spoofing sets as they have a common repeatable signature in the files tested and were not observed in any of the live field data processed. On the other hand, the metric G_{AGC}, sees a big impact as soon as the spoofing attack starts, and the metric surpasses the threshold for the whole duration of the test. In these figures we can understand the importance of having a control on the gain of the AGC. As can be noticed, it is an simple and effective way to detect over-powered spoofing scenarios.

Scenario ds3

Figs. 5.14 and 5.15 show the results for G_{AGC} and Δ, respectively, for the spoofed dataset ds3. We observe how in this scenario, metric Δ is more powerful and it is able to detect the asymmetries formed in the correlation function once the spoofer starts separating the two signals, between 100 and 250 seconds. After that, the two signals are completely separated and this version of the SQM is not able to detect asymmetries anymore, following the expected transient indicator trend. For metric G_{AGC}, we see that it also detects the spoofing presence, but the impact is not as high as observed for ds2. A spoofer with a lower power advantage, and a receiver

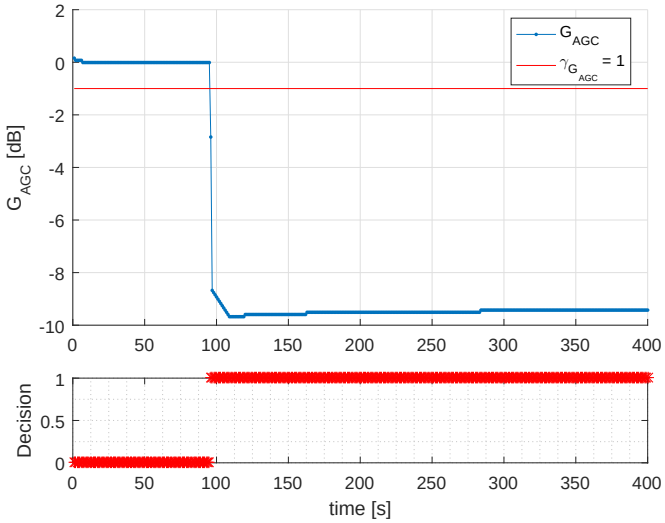

Fig. 5.12 Trend for G_{AGC}, along with the decision taken for the TEXBAT scenario ds2. The G_{AGC} is able to detect the spoofer presence after 95 seconds which is when the spoofing started for this scenario

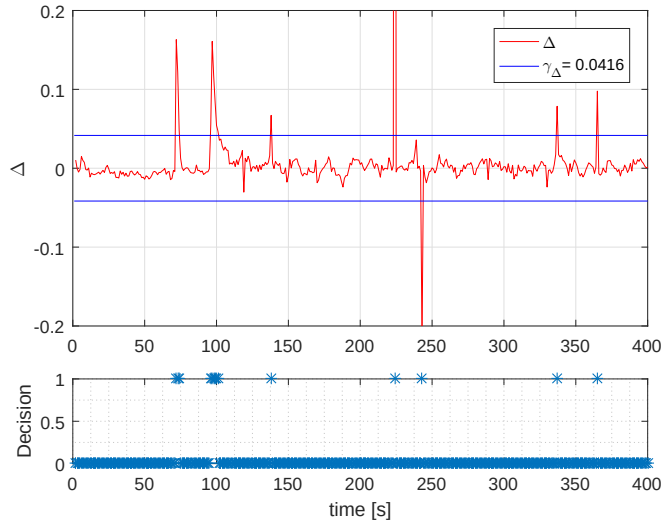

Fig. 5.13 Trend for Δ, along with the decision taken for the TEXBAT scenario ds2. The metric Δ struggles with detecting the distortions in the correlation function and a handful of detections are present

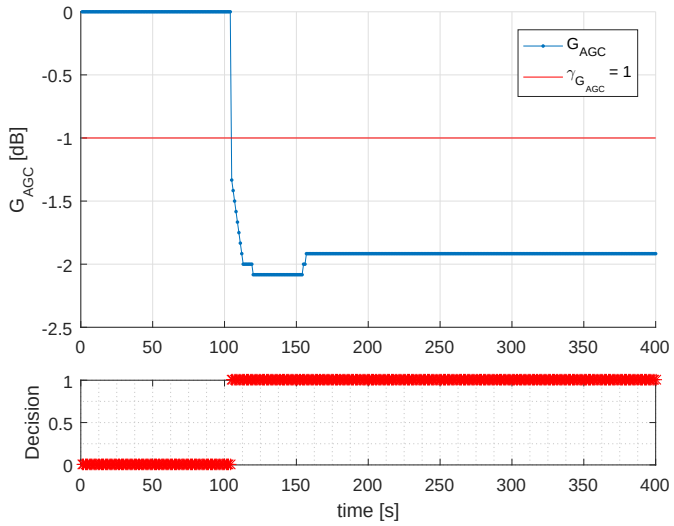

Fig. 5.14 Trend for G_{AGC}, along with the decision taken for TEXBAT scenario ds3. Metric G_{AGC} is able to detect the presence of the spoofer after 110 seconds which is when the spoofing started

with enlarged thresholds, could reduce the impact on the AGC to levels below the threshold. Such a configuration would greatly increase the impact on the SQM metric Δ, making it easily detectable that way.

The metrics Δ and G_{AGC} are very powerful tools that can be used for spoofing detection with success as shown in this Section. Observing distortions of the correlation shape indicates that another *"GNSS-like"* signal is present and that the solution cannot be trusted. The metric Δ is especially powerful for detecting attacks where the spoofer does not have a significant power advantage over the satellite signals. Some false alarms may be generated by multipath signals in the environment, but this problem can be addressed as proposed in Section 4 and [55], where the multi-dimensional SQM metric is proposed, or by planned localization of the receiver's environment, if it is placed in a static position. In the following Sections we shift away our focus from the SQM metrics and we focus on reducing false alarms in the overpowered cases, detected by G_{AGC}.

We hypothesize the false alarms triggered in the TEXBAT datasets ds2 and ds3 around 50 seconds are an artifact of the creation of the spoofing sets as they have a common repeatable signature between the different files and are not observed in the clean datasets.

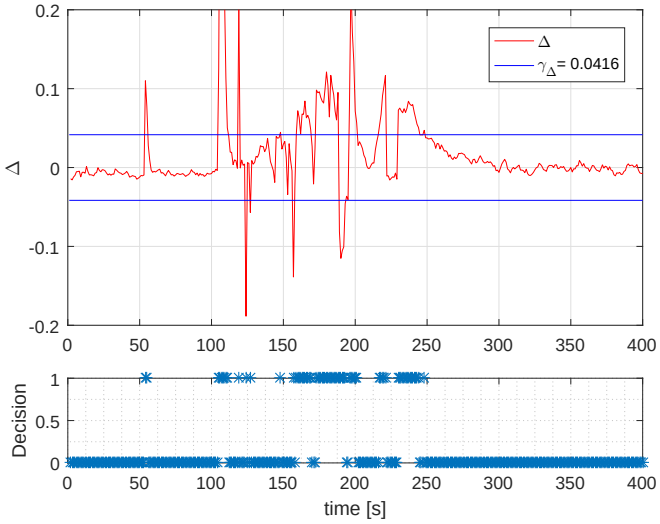

Fig. 5.15 Trend for Δ, along with the decision taken for TEXBAT scenario ds3. The metric Δ sees a big impact once the attack starts, after 110 seconds and it is able to detect distortions until 250 seconds

5.4 RFI and Spoofing

Examining the power measurements via the AGC monitoring and metric G_{AGC} provides hints that additional signals are present in the receiver bandwidth and that the receiver is being affected by RFI. This RFI can be produced by intentional sources like spoofers or jammers or by unintentional sources like TV antennas, or some near band spill-over. The impact that the interference has on the AGC is very similar from one source to the other, given that they all add additional power within the band.

Jamming attacks have been a concern for GNSS users for a long time, given the low power of the received signal as observed in Fig. 5.1. Multiple publications highlight the impact of jammers on GNSS receivers [12, 23], and some real scenarios have been observed. As a known example, in 2011, a truck driver using a personal privacy device (PPD), with the goal of disrupting the signal of the GPS tracker installed in his truck, heavily jammed GNSS signals, used in Newark airport in the USA, everytime it passed close by [72]. Other examples have resulted in coastal guard warnings, declaring neglected GNSS usage, due to interference events [30, 28, 68]. All these situations, even if not necessarily maliciously envisioned, make the

Fig. 5.16 Location of HNL station in Honolulu, Hawaii

reliable use of GNSS signals, especially for critical application, a difficult task. An example was encountered during the analysis of the WAAS station data.

In order to better illustrate the point, we show, in Fig. 5.16, the location of the Honolulu (HNL) WAAS station. We can observe that the location is in a relatively clear position, with the sea close by and no big urban areas surrounding it. These characteristics are reflected in the clean trend of G_{AGC}, as observed in Fig. 5.4.

Let's now look at the position of WAAS station ZMA, in Miami, Florida. We can see that it is positioned inside a deep urban area, with many buildings and major highways nearby. If we now observe the trend of G_{AGC} for the ZMA station, shown in Fig. 5.18, we can see how RFI is affecting the measurements of G_{AGC}. In this case, the interference is likely coming from nearby radio frequency sources. Similar trends are observed for other stations near major highways, like ZBW, where localized jamming may be a problem.

These RFI effects on the receiver are similar to the ones generated by an overpowered spoofing attack. They both affect the metric G_{AGC}, while the SQM metric does not suffer from any considerable distortions. In order to lower the false alarms produced by the RFI, in Section 5.4.1 we describe a possible approach to discriminate between the two events.

Fig. 5.17 Location of ZMA station in Miami, Florida

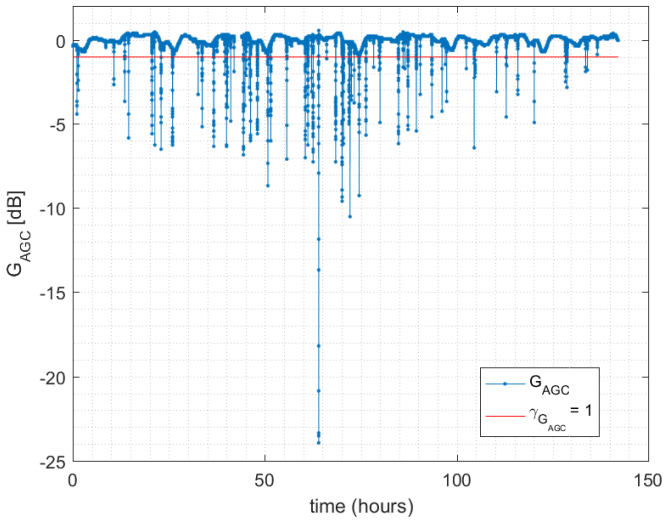

Fig. 5.18 Trend for G_{AGC} in WAAS station ZMA, in Miami, Florida. The station is affected by RFI interference coming from the surrounding urban areas.

5.4.1 Observation of the C/N_0 and the AGC

In order to understand how to discriminate between RFI and spoofing attacks, we first need to highlight the way the interference is generated and its goal. In the case of a spoofer, the attacker wants to send a counterfeit signal that is aligned with the GNSS satellite's signals. This new signal will effectively increase the power of the carrier signal when both the spoofing signal and the satellite's signals are aligned.

During RFI events, such as jamming, a signal that is *not* consistent with the satellite signal is added into the bandwidth. Basically, noise is added to the GNSS band, distorting or neglecting its usage. Due to the increased power observed in the band, the AGC lowers its gain value. For RFI events, if we observe the C/N_0 value, a drop will be observed due to the increased level of the noise floor, N_0. On the other hand, during a spoofing attack, due to the addition on the carrier measurement, we will naturally see a higher value of C/N_0. This increment is due to the alignment in carrier frequency that the spoofer and satellite signals have. This is especially true in the overpowered spoofing attacks, that are the focus of this Section.

To better illustrate the effects of the jamming and the spoofing attacks, in Fig. 5.19 we show an example of the effect of the RFI event in the curve of power versus frequency. In the Figure we observe how, for the example jamming event in A, the noise level is raised by 20 dBm, burying the satellite signal deeper into the noise. On the other hand, for the example spoofing event in B, we observe how the total power of the signal seen by the receiver is raised by 20 dBm, due to the presence of the spoofing signal on top of the one coming from the satellite. This effect results in a higher C/N_0 being observed by the receiver.

A more advanced spoofer may be able to maintain the C/N_0 value constant or even decrease it by adding additional noise to the band and bury the satellite signal under the noise. Nevertheless, the metric G_{AGC} will notice that larger amounts of power are being inserted, and that C/N_0 is not decreasing with the expected trend. If we characterize the AGC vs C/N_0 response of the receiver's front-end to RFI events, we are able to draw a threshold that will allow us to discriminate between RFI events and spoofing attacks.

Fig. 5.20 shows an example of how the G_{AGC} value compares to the C/N_0 for both interference (on the left) and spoofing attack (on the right). Having a complete

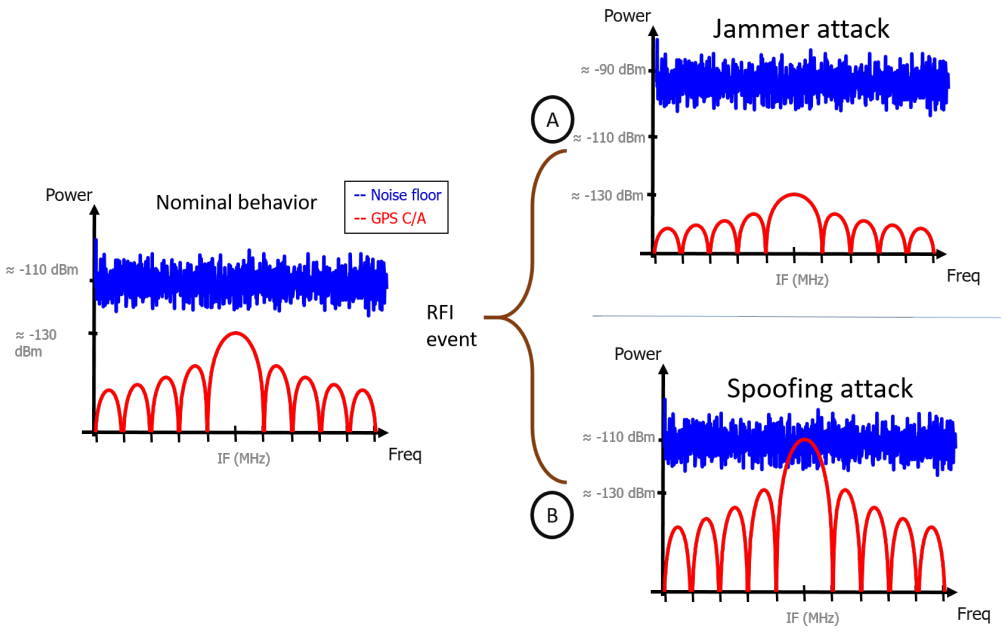

Fig. 5.19 Examples of the effects of the Jamming and Spoofing attacks in the frequency domain. In (A), we observe how the noise floor is raise for the jamming attack, while in (B) the total signal power is increased, due to the presence of the spoofing signal

knowledge of the response of the RF front-end of our receiver, we could use this information for differentiating between the two types of interference.

In the case that we plot the curve of the AGC variations versus the C/N_0 difference, we can distinguish the two effects by separating the space into two regions as observed in Fig. 5.21. In the Figure we plotted, in small dots, the trend of three different WAAS stations, two of which contain presumed interference (ZMA and ZBW), while in stars we plotted the trend of TEXBAT scenario ds2.

We can observe how in this example the spoofing scenario falls to the right-hand-side of the red line, while the jamming cases fall to the left of it. Given this initial result we wanted to validate this behavior and we performed controlled interference tests in the lab in order to assess the thresholds and limitations of the technique. These results are presented in Section 5.4.2.

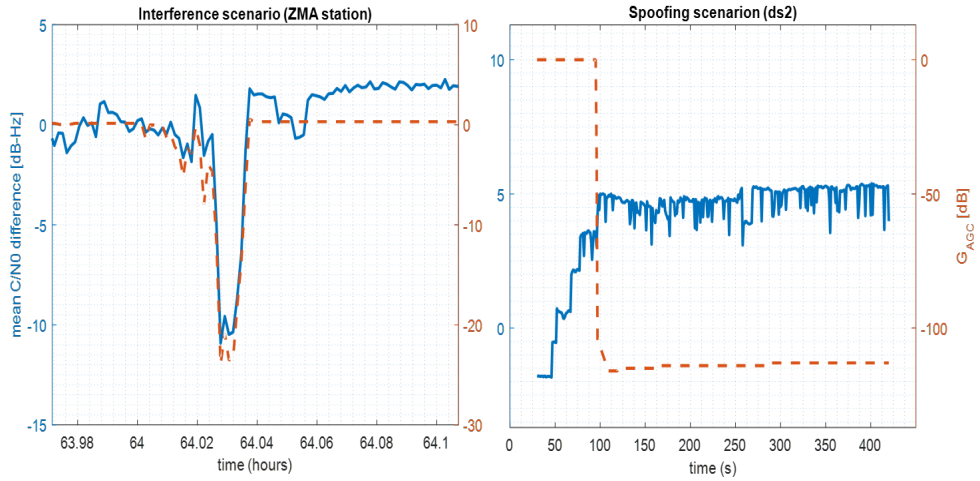

Fig. 5.20 Examples for C/N_0 difference (in blue) and AGC difference (in brown), for both interference and spoofing attacks

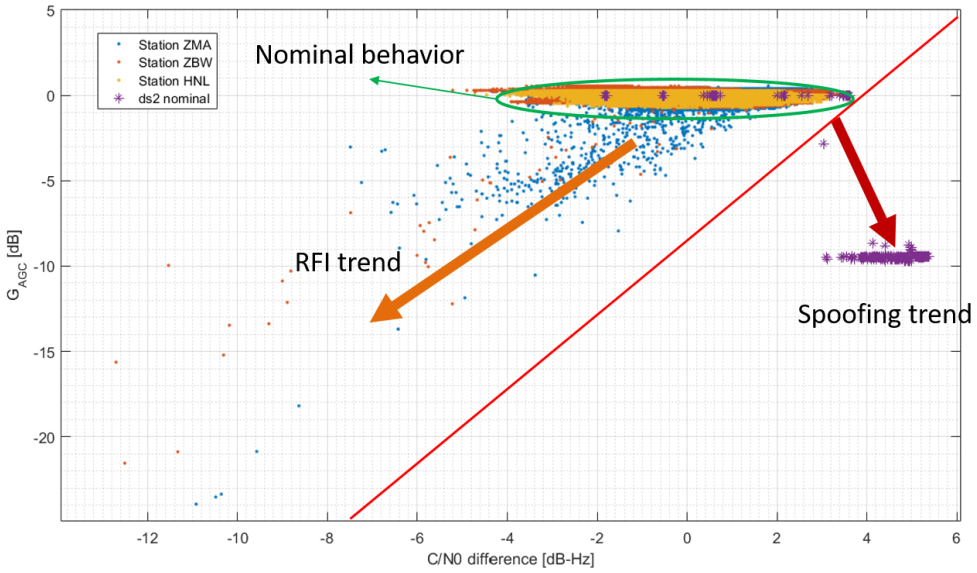

Fig. 5.21 Examples for G_{AGC} vs C/N_0 difference. To the left of the red line we can observe the trends for different WAAS stations, while to the right we observe the spoofing scenario ds2. The Nominal behavior circled points are obtained when no RFI nor spoofer are present. Basically the C/N_0 fluctuates around $+/-4$ dBHz from the mean value and the G_{AGC} fluctuates around $+/-2dB$ from the mean.

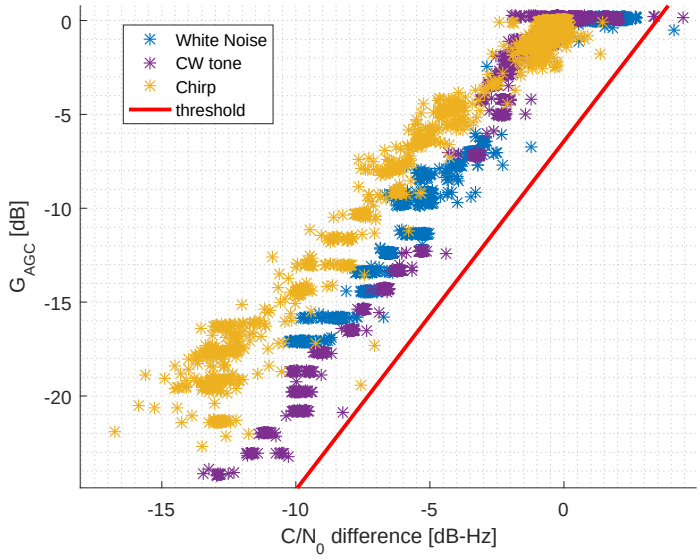

Fig. 5.22 G_{AGC} vs C/N_0 trend for the receiver affected by different types of interference and analytic threshold obtained (in red)

5.4.2 Controlled environment interference

In order to assess the feasibility of the technique and the thresholds for distinguishing between spoofing attacks and RFI, we need to understand the behavior of the receiver when it is affected by interference. In order to do this, we injected different types of interference to the receiver in a controlled environment and observed its characteristic response. Three types of jammer where used, one injecting only white noise, one injecting a single continuous wave (CW) tone centered in the GPS L1 frequency and the chirp, which modifies the frequency of the single tone quickly in order to disrupt wider band of the receiver. These jammers were transmitted via cable connection to the receiver and their power was modified by means of variable attenuators. In Fig. 5.22, we can observe the trend of G_{AGC} vs C/N_0, for the three different types of interference transmitted to the receiver.

Using these interference results we can draw a threshold that contains all interference cases to the left, as shown in red in Fig. 5.22 and use that threshold to identify the cases of spoofing and interference that were shown previously.

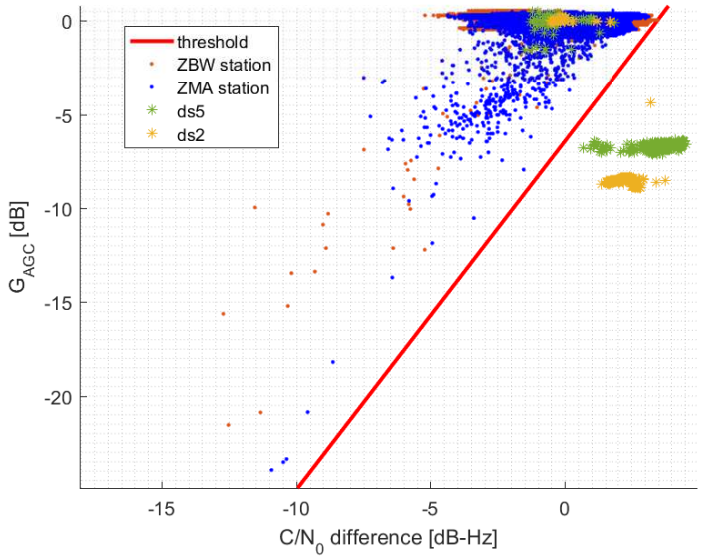

Fig. 5.23 G_{AGC} vs C/N_0 trend for the receiver, obtained by means of the different datsets for the WAAS stations and the TEXBAT datasets. In small dots the presumed RFI results are shown for stations ZBW and ZMA, while in stars the overpowered spoofing datasets (ds2 and ds5) are shown. The analytic threshold obtained from Fig. 5.22 is shown (red). We can observe how the threshold distinguish between the two types of interference

Fig. 5.23 shows the results when using the threshold obtained from the interference in a controlled environment (5.22), to distinguish between spoofing and jamming signals.

To the left of the threshold, we observe the results of the two WAAS stations that are affected by what is assumed to be interference, while on the right of the threshold the dots correspond to two overpowered spoofer cases, scenarios ds2 and ds5, re-transmitted to the receiver.

From the results in Fig. 5.23 we observe that with the characterization and knowledge of the behavior of the AGC inside the receiver, we are able to discriminate between a jamming interference that aims at disrupting the tracking of the signal and a spoofing attack that aims at taking control of the receiver. It is also worth flagging that a check on temporal dynamics could also be effective when trying to distinguish between the two phenomena, if we assume that the jamming device will only be in range of the station for a few seconds, while the spoofer would need a significantly longer presence in order to complete its goal.

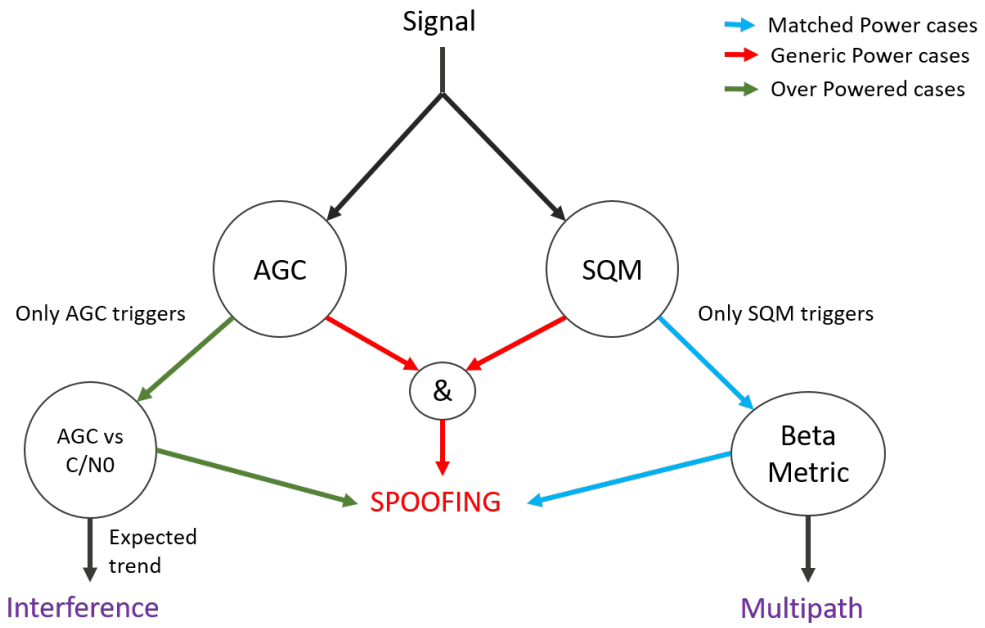

Fig. 5.24 Flow chart of the spoofing detection algorithm, containing the different techniques and discussions done during Part II of this thesis.

5.5 Combined spoofing detection algorithm

In this Chapter we have shown the effectiveness of combining an SQM metric, for correlation distortion observation, with AGC measurements, for interference power monitoring. We also showed a way of computing thresholds for the nominal behavior of commercial receivers, by means of multiple data collections. Finally, we addressed the question of false alarms in spoofing detection, when using the AGC metric, due to the presence of jamming events.

To reduce these false alarms, we proposed a technique where the joint observation of the C/N_0 and G_{AGC} values, given that this relationship will be different between the case of a spoofing attack and the case of a jamming event.

To conclude Part II of the thesis, which focused on the development of spoofing detection algorithms and on the processes to lower the probabilities of false alarms, due to other external events, we present, in Fig. 5.24, the combined flow chart of the algorithm that a GNSS receiver with the need to have an anti-spoofing module, could implement for detection purposes.

In the flow chart, we observe how the signal coming from the antenna, in black, is observed simultaneously by both the AGC metric and the SQM, looking for spoofing presence. If both metrics trigger at the same time, a spoofing attack is declared to be present in the signal (in red). This detection is done without fear of false alarms due to the presence of other external events.

If only the AGC metric is triggered, the signal could contain an overpowered spoofing attack or a jamming event. In order to discriminate between the two possibilities, we use the AGC vs C/N_0 trend, as discussed in Section 5.4, and the distinction between interference (in black) and spoofing (in green) can be done. In the case of a heavy multipath environment and signal blockage, i.e. when the user enters indoor, the C/N_0 will likely drop, but the AGC will not be affected, or at maximum will increase its level, due to the lack of presence of satellite signals. This means that the check of AGC vs C/N_0 would not be done.

On the other hand, if only the SQM triggers when the signal is observed, it means that the signal could be affected by a matched-power spoofing attack or by multipath signal. In order to discriminate between them, we use a multidimensional ratio metric, as the one proposed in Chapter 4. The distinction between multipath signal (in black) and spoofing presence (in blue) is performed.

Combining all of these different methods, we obtain a reliable way of detecting the spoofing presence, with a low complexity level, low probability of false alarms, and fast response to the events, that could be implemented in commercial GNSS receivers. This combination of techniques is able to detect most configurations of the spoofing attacks, thus minimizing the degrees of freedom that the spoofer has.

Finally, it is important to emphasize that with the proposed combined algorithm, only *detection* of the spoofing event is possible. The algorithms are developed to detect the spoofing presence as fast and reliably as possible, but once detected, they cannot prevent the spoofer from gaining control of the receiver. The goal is to raise a warning, and for the receiver to stop trusting its GNSS navigation solutions, given that they are being compromised.

In Part III of this thesis, we focus on techniques that allow for detection of the spoofer presence *and* that are able to mitigate the spoofer effects. As will be seen in Chapters 6 and 7, these mitigation techniques are more complex and require a higher computational load.

Part III

Spoofing Mitigation

Chapter 6

Time jumper algorithm

Mitigation of spoofing effects is a cumbersome topic. For many applications, especially safety critical ones, detection of the spoofing attack would be enough, even if that means shutting down the GNSS receiver usage. These applications may not accept navigation solutions using mitigated signals, because they are, by nature, less reliable than the solutions obtained with clean satellite signals. Other applications may accept mitigated signals and maintain GNSS usage, especially if they do not need very high accuracy out of the GNSS solution.

This situation put spoofing mitigation techniques in an interesting position, where they may be desired, but not always necessary. As a matter of fact, not many receiver based spoofing mitigation techniques have been investigated. Usually, spoofing mitigation in GNSS is achieved by means of antenna arrays or specialized antenna hardware, that can steer the nulls of the antenna, to reduce the spoofing signal presence in the receiver [64, 60, 19, 56].

In this Chapter and in Chapter 7, we will present two different signal processing alternatives for spoofing mitigation. These techniques are not only able to detect the spoofing presence, but also use knowledge of the spoofing signal characteristics to mitigate its effects.

In this Chapter, a novel technique is presented, referred as Time Jumper (TJ) algorithm[1]. The algorithm has the goal of estimating the parameters of the spoofing and satellite signals, present in the same correlation space, by means of linear

[1]The TJ algorithm was developed in collaboration with fellow PhD. student Mattia Berardo and professor Letizia Lo Presti.

regression algorithm. Once the parameters are estimated, the TJ algorithm modifies the delay of the DLL, in order to obtain mitigated pseudoranges.

The TJ algorithm focuses on the matched-power intermediate spoofing attack (see Section 2.4.2), and its feasibility is demonstrated, in particular, on dynamic scenarios and applications. For validation we used the TEXBAT datasets [38] in order to obtain baseline results. The TJ algorithm, by itself, is not conceived as a defense against the over-powered types of spoofing attacks, and these would need to be detected by a power measurement observation, like proposed in Chapter 5.

By means of the TJ algorithm, the receiver is able to maintain availability of the GNSS usage and obtain trusted positions, using measurements obtained from the mitigated signal. Due to the presence of vestigial spoofing signals, the accuracy of mitigated positions is slightly degraded with respect to the position obtained using clean signals.

This Chapter is based on the work presented in [10].

6.1 The time jumper principle

Before detailing the multiple elements used by the TJ algorithm, we want to explain its basic working principle. The flow chart of the algorithm describing the TJ functionalities is depicted in Fig. 6.1, where three main sections of the algorithm can be identified, the detection part, the PVT part, and the jumping part.

The detection part is in charge of revealing the spoofing presence and of its exclusion from the navigation solution. The detection is based on a Signal Quality Index (SQI), able to measure the difference between the received signal and the ideal one. The SQI is defined by using the correlation function between the received signal and the local code replica, and other signal information, such as C/N_0. The SQI value is the metric used for the detection and exclusion of the faulty satellites from the PVT computation. A detailed description of the detection part is presented in Section 6.2.

The PVT part, described in Section 6.3, regards the use of Kalman filters in the PVT computation. There are two different Kalman filters used in the algorithm, one used when enough unspoofed channels are available to compute a trusted solution

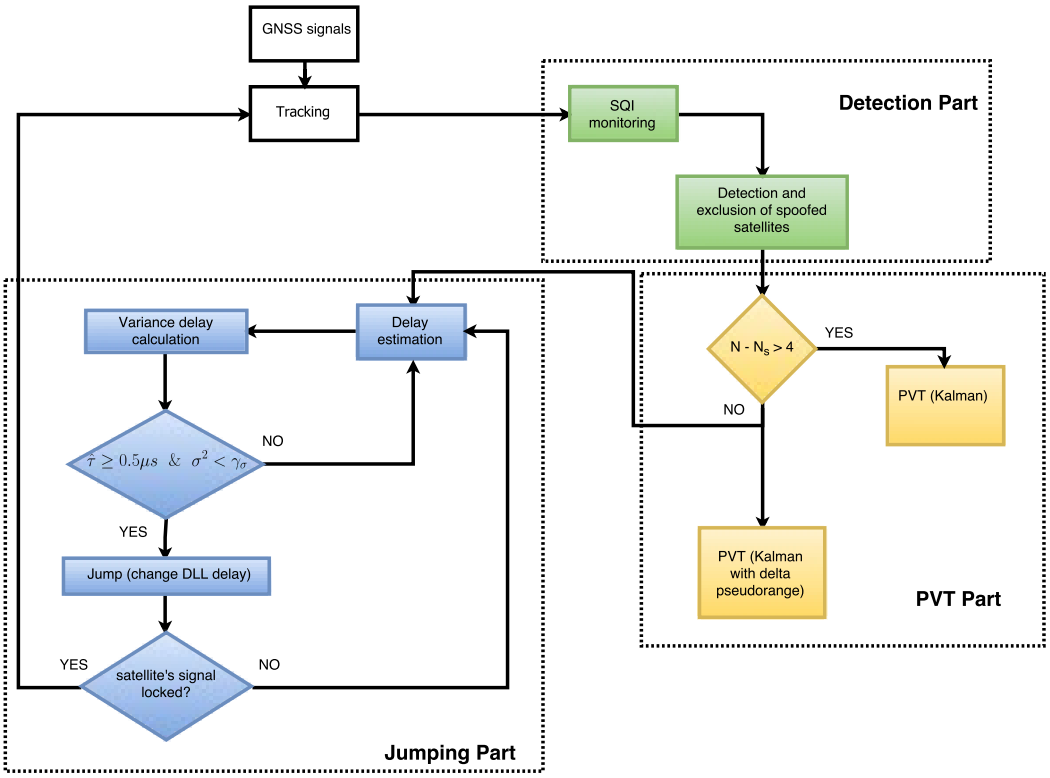

Fig. 6.1 Flow chart of the functioning principle of Time Jumper algorithm

and the second one when a solution cannot be found using only the unspoofed channels, as declared by the detection metric.

Finally, in Section 6.4 we discuss the jumping part that is in charge of unlocking the channel's DLL from the control of the spoofer and lock it to the authentic signal. During this procedure, we first estimate the relative delay between the spoofing and the satellite signal by observing the correlation space. The delay of the DLL is then modified by using the estimated delay difference between the spoofer and the satellite signals and this procedure is named the delay jump. After the jump, appropriate checks to control if the jump was performed correctly are done.

The detection strategy and the PVT computation are based on previous research and known results. The jumping procedure is the major novel contribution of this work.

6.2 Detection part

The detector, used in the scheme of Fig 6.1, is a modified version of the Multipath Distance Detector (MPDD) algorithm introduced in [8, 9]. This algorithm works at the correlation level and it is able to detect distortions of the correlation function. Since multipath and spoofing signals may create similar distortions in the correlation function as we have seen previously, it is possible to use a variant of the MPDD algorithm to detect spoofing signals. In spite of the similarities in the distortion of various impairments situations, there are subtle differences that can be considered when differentiating between them. As an example, a spoofing attack is expected to create a single additional ray apart from the LOS signal, while in multipath environments many different rays can be mixed with the LOS signals.

The MPDD is based on the use of Linear Adaptive Filters (LAFs). In this work we substitute the LAFs by a Least Absolute Shrinkage and Selection Operator (LASSO) [80], which provides, in the framework of spoofing detection, better detection capabilities. Furthermore, the algorithm is improved by the introduction of a quality index that will be used to exclude low quality signals.

6.2.1 From linear adaptive filter to LASSO

The idea described in [8] is to use LAF to decompose the correlation function of the incoming signal in a weighted sum of delayed, and ideal, correlation functions in order to detect multipath reflections of the LOS in the correlation domain. In the definition there is no limit on the number of reflections accepted in the decomposition. In the LAF theory, a generic signal is modeled as

$$d[n] = y[n] + n_0[n] \tag{6.1}$$

where $n_0[n]$ is a noise sequence and $y[n]$ is a signal defined as the output of a Finite Impulse Response (FIR) filter:

$$y[n] = \sum_{k=0}^{M-1} w_k^* u[n-k] \tag{6.2}$$

according to (6.2), $y[n]$ is written as a summation of M delayed and weighted replicas of a basis input signal $u[n]$. The filter length M depends on how many delayed replica

signals are used to approximate $d[n]$. In [8], $u[n]$ is the ideal correlation function of a GPS L1 C/A signal and $d[n]$ is the measured correlation function, between the local code and the incoming one, averaged in a time window. In order to limit the effects of the noise in the receiver, the measured correlations are collected in time and then averaged before applying the LAF decomposition.

The taps of the filter w_k are considered as the unknowns of the system and they are computed by minimizing the residual error $e[n] = y[n] - d[n]$. This minimization problem can be written in matrix form as:

$$\min_{w} \quad ||\mathbf{U}\mathbf{w} - \mathbf{d}||_2^2 \tag{6.3}$$

where $\mathbf{U} \in \mathbb{R}^{N \times M}$ contains M delayed ideal correlations and $\mathbf{d} \in \mathbb{R}^{N \times 1}$ is a vector of measured correlation points. Equation (6.3) is a typical least square minimization problem, solved as $\hat{\mathbf{w}} = (\mathbf{U^H U})^{-1}\mathbf{U^H d}$. The vector $\hat{\mathbf{w}} \in \mathbb{R}^{M \times 1}$, containing the taps of the linear filter, is used to characterize the presence of external signals. The value of the central tap, weights the ideal correlation with zero delay and is related to the LOS signal, while the values of other taps are linked to the presence of possible multipath signals and noise.

The rule chosen here to detect the presence of distortions is the same used in [8]. We compute the square distances between $\hat{\mathbf{w}}$ and a set of sample vectors $\mathbf{w}_p \in V$, so

$$E_{LOS} = \|\hat{\mathbf{w}} - \mathbf{w}_{LOS}\|^2$$

and

$$E_p = \|\hat{\mathbf{w}} - \mathbf{w}_p\|^2 \qquad p = 1, 2, ...N_V$$

where N_V is the number of vectors in the set V. Observing these distances we are able to decide if the correlation function is distorted. Following, we decide that anomalies are present if

$$\min_{p=1,2,...N_V} E_p \neq E_{LOS} \tag{6.4}$$

The vectors in V take into account different scenarios, including LOS condition with no extra signals ($\mathbf{w}_{LOS}$) and multipath with multiple rays ($\mathbf{w}_p$). The vectors in V identify all the possible binary combinations of the length of the filter M, where the coefficients are multiplied by an additional parameter α, such as $-1 < \alpha < 1$, and

$\alpha \neq 0$. This parameter α adjusts the amplitude of the considered delayed replicas, with the exception of the central one [8].

In spoofing attack scenarios the problem is similar. Distortions are present in the correlation function, but only one additional signal is expected. The minimization problem, expressed by (6.3), can be modified to take into account the fact that, for spoofing attacks, we are looking for a single additional signal. The new formulation of the problem has to guarantee a limited number of non-zero components of the vector $\mathbf{w}$, and this is possible by adding a constraint so that:

$$\begin{aligned} \min_{w} \quad & ||\mathbf{U}\mathbf{w} - \mathbf{d}||_2^2 \\ \text{subject to} \quad & ||\mathbf{w}||_0 < \varepsilon \end{aligned} \tag{6.5}$$

where $||\mathbf{w}||_0$ is the pseudo ℓ_0 norm defined as

$$||\mathbf{w}||_0 = |\text{supp}(\mathbf{w})|$$

and where $|\cdot|$ is the cardinality of the set $\text{supp}(\mathbf{w}) = \{\mathbf{w} : w_i \neq 0\}$. The integer number ε, limits the ℓ_0 norm of the vector $\mathbf{w}$, so s indicates the maximum number of non-zero components. In our case, a possible choice is to select $\varepsilon = 2$, in order to represent only the LOS and the spoofed signal. Nevertheless, the replicas in $\mathbf{U}$ have fixed delay values while the LOS and the spoofing signals may fall between two replicas, as shown in Fig. 6.2 and this leads to $\varepsilon \geq 4$. To limit the complexity of the overall representation we chose $\varepsilon = 6$.

The problem in (6.5), an ℓ_0 *constrained least square*, is a non-convex and NP-hard problem [14]. In order to solve it, a possible approach could be to use greedy algorithms like *iterative hard thresholding* [11] or, as an alternative, we relax ℓ_0 and solve an approximated version of the problem, using the ℓ_1 norm [73].

For the TJ algorithm, we are more interested in highlighting the presence of components than approximating the input signal in the best possible way. The convex problem

$$\min_{w} ||\mathbf{U}\mathbf{w} - \mathbf{d}||_2^2 \tag{6.6}$$

$$\text{subject to } ||\mathbf{w}||_1 < \varepsilon \tag{6.7}$$

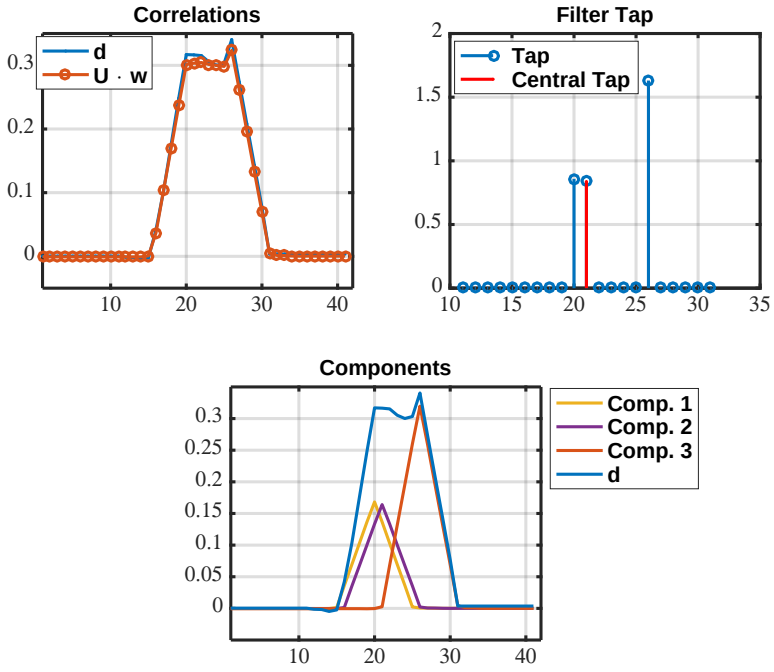

Fig. 6.2 Example of measured correlation **d** and its approximation, tap of the filter (**w**) and weighted decomposition

is referred to as the LASSO problem [80, 31]. Given the nature of the constraints, the LASSO tends to produce coefficients that are exactly zero [77] and this behavior fits well with the desired representation for the TJ algorithm. In our case, we have $\mathbf{w} \in \mathbb{R}^{M\times 1}$ with $M > \varepsilon$, so the solution will have at most ε components different than zero. In Fig. 6.2 we can observe an example of the LASSO, applied to decompose input correlation **d** and approximate it as $\mathbf{y} = \mathbf{U}\mathbf{w}$. The vector **w**, obtained by the LASSO solution, is used for detection purposes. Vector **w** will also be used in Section 6.4.1 to estimate the relative delay between LOS and spoofing signals.

6.2.2 Signal Quality Index

Another element of the proposed method is the introduction of an index that evaluates the quality of the signal. The output provided by the detector in (6.4), is a hard detection (Yes/No) about the presence of correlation anomalies. Even a single and small coefficient different from zero in **w**, would be interpreted as a positive detection.

However, it could be interesting to have a soft decision to better discriminate cases in which the signal is heavily degraded from the actual presence of a spoofer.

The idea is to continuously monitor the quality of the received signal during the computation of the navigation solution. To achieve this goal we introduce the $\mathrm{SQI}(t_n)$ that describes the quality of the signal, where t_n are the time instants of the PVT computation. This quality index is defined as

$$\mathrm{SQI}(t_n) = \frac{1}{\lambda} \sum_{k=1}^{N} m(t_{n-1} + kT_a) d(t_{n-1} + kT_a) f\left[\frac{C}{N_0}(t_{n-1} + kT_a)\right] \tag{6.8}$$

and takes into consideration the MPDD output and other parameters such as the C/N_0 and the distance between the weight vector $\hat{\mathbf{w}}$ and the theoretical LOS vector. The components of (6.8) are:

- T_a is the duration of the average time window.
- N is the number of average time windows between two time instants (t_n and t_{n-1}). For example, between two PVT computations with a rate of 1 Hz and a average time window of $T_a = 100$ ms, $N = 10$.
- $m(\cdot)$ represents the MPDD output,

$$m(kT_a) = \begin{cases} -1, & \text{if } E_{min} = \min E_p = E_{LOS} \\ +1, & \text{if } E_{min} = \min E_p \neq E_{LOS} \end{cases}$$

  for $p = 1, 2, ...N_V$, meaning that -1 indicates presence of only the LOS signal, while $+1$ indicates the presence of distortions.

- The function $d(\cdot)$ measures the ratio E_{min}/E_{LOS}, and it is defined as

$$d(kT_a) = \begin{cases} E_{min}/E_{LOS}, & \text{if } E_{min} \neq E_{LOS} \\ 1, & \text{if } E_{min} = E_{LOS} \end{cases}$$

  therefore, $d(\cdot)$ is a ratio between distances, and it gives a relative measure of how different is $\hat{\mathbf{w}}$ with respect to $\mathbf{w_{LOS}}$.

- $f(\cdot)$ takes into account the C/N_0 value, and is defined as:

$$f\left[\frac{C}{N_0}(kT_a)\right] = \frac{1+\frac{2}{\pi}\arctan\left[\frac{C}{N_0}(kT_a)-a\right]}{2}$$

  The C/N_0 gives a measure of the reliability of the output of the MPDD. As a matter of fact, if C/N_0 is high, it means that the decision about the presence of distortions is more reliable than with a C/N_0 closer to a. The value a is the minimum operative C/N_0 considered by the algorithm. We chose an arctangent function to take into account a saturation effect in case of very high or very low C/N_0.

  It is important to choose a C/N_0 estimator according to our problem. In [26] a comparison between five well-known methods to estimate the C/N_0 is made and based on it, we chose to use an estimator with low computational complexity, the *Signal-to-Noise Variance* (SNV), that is based on the first absolute moment and the second moment of the signal samples [26].

- λ is a normalization factor to obtain $\mathrm{SQI}(t_n) \in (0,1]$. The parameter λ is set as the sum of the maximum values that can be obtained at each discrete time k for the functions $m(\cdot)$, $d(\cdot)$ and $f(\cdot)$.

In Figs. 6.3 and 6.4, two examples of MPDD binary outputs, SQI results and C/N_0 trends are depicted. In these Figures we observe how the spoofing signal affects the correlation once the SQI value drops considerably, after 180 s in Fig. 6.3 and after 110 s in Fig. 6.4.

6.2.3 Exclusion rule

A spoofing attack is able to modify the delay of each satellite signal at different times and with different trends, based on the desired position where it wants to *move* the receiver. The mitigation of the dangerous spoofing effects cannot be based only on SQI, given that a low SQI does not necessarily indicates that the receiver is under spoofing attack. In case of the TJ algorithm, we declare the spoofing presence by combining the information given by the SQI and the duration of the event.

In order to make the discrimination, we defined a threshold γ_{SQI} and we compare the obtained value of SQI against it for anomalies detection. Afterwards, we

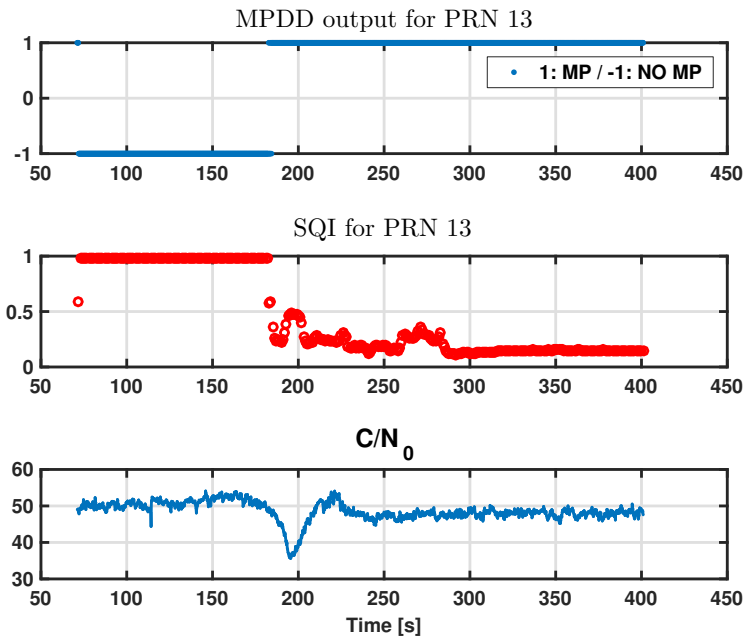

Fig. 6.3 Example of detection results for spoofing scenario. The attack is detected after 180 s, when the spoofer tries to modify the true delay computed by the receiver.

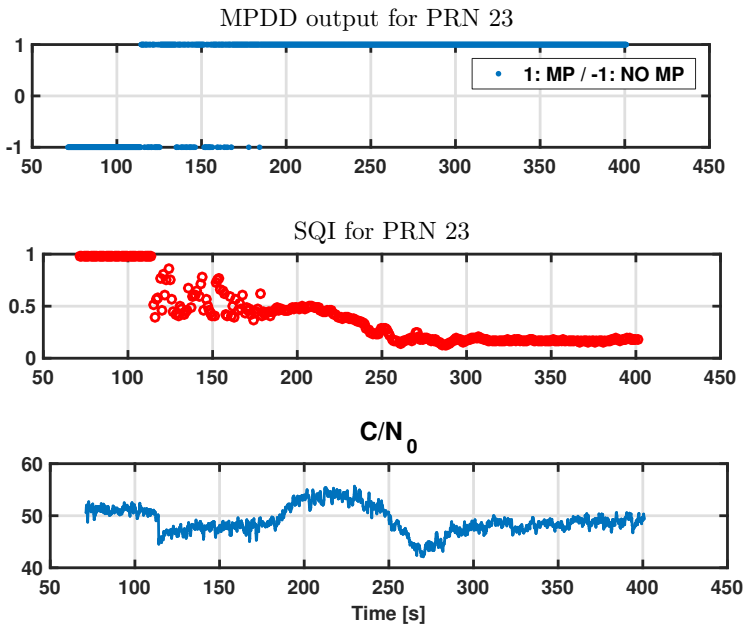

Fig. 6.4 Example of detection results for spoofing scenario. The attack is detected after 110 s, when the spoofer tries to change the true delay computed by the receiver.

discriminate between a multipath and a spoofing signal by observing the duration of the event. If the SQI is below the threshold, for a certain amount of time T_{SQI}, we declare the spoofing presence. The decision d_T, for channel i, can be summed up as:

$$d_T^{(i)}(t_n) = \sum_{t_x=t_n-N_d+1}^{t_n} SQI^{(i)}(t_x) < \gamma_{\text{SQI}} \tag{6.9}$$

and the decision is taken for:

$$d_T^i(t_n) \begin{cases} = N_d \longrightarrow \text{spoofed} \\ > 0 \text{ and } < N_d \longrightarrow \text{distorted} \\ = 0 \longrightarrow \text{no distortion} \end{cases} \tag{6.10}$$

where N_d is equal to the number of values of the SQI, in the time interval between two decisions T_{SQI}. After analyzing the trend of the SQI and taking into account the spoofing dynamics, we defined a heuristic value for $\gamma_{\text{SQI}} = 0.6$ and $T_{\text{SQI}} = 10$ s. These thresholds should be adjusted based on the expected behavior of the spoofing and the receiver's environment.

If a channel is declared as *spoofed*, it is excluded from the navigation solution. Once a channel is excluded, the remaining healthy ones are used to compute the PVT solution by means of a classical Kalman filter. The algorithm will continuously monitor the channels and exclude the impaired ones. If there are less than four channels declared healthy, the algorithm will switch to the Kalman filter version used under spoofing attack, and the solution will be computed using only Doppler measurements as it will be explained in Section 6.3. At the end of the procedure, a number N_s of satellites will be excluded, where $0 < N_s \leq N_T$ and N_T is the total number of satellites.

6.3 Kalman filter

The Kalman filter was first introduced in 1960 by Dr. R. E. Kalman [44], and the usage of the Kalman filter in a GNSS receiver is thoroughly described in [45] and [15]. It is currently used for many different applications as an estimation and filtering technique. The Kalman filter combines knowledge of the system dynamics and

knowledge of the statistical errors of the system, in order to estimate the state of the system at a given time.

In the TJ algorithm, when $N_T - N_s \geq 4$, N_s satellites are excluded from the solution based on the Kalman filter. This filter only uses the non-spoofed code and Doppler measurements. Otherwise, when $N_T - N_s < 4$, all the N_T Doppler measurements are used by the Kalman filter. This dual approach provides an overall better performance, in terms of continuity and accuracy, as it will be shown in Section 6.5.

Both discrete Kalman filters architectures are based on the description presented in [45, 15] and the notation used hereafter, referring to time t_k, is:

- $\mathbf{x_k}$ is the process state vector
- $\phi_\mathbf{k}$ is the state transition matrix relating $\mathbf{x_k}$ to $\mathbf{x_{k+1}}$ in the absence of a forcing function
- $\mathbf{z_k}$ is the measurement vector
- $\mathbf{H_k}$ is the matrix giving the ideal connection between the measurement and the state vector
- $\mathbf{v_k}$ is the vector containing the measurement error, which is assumed white with known covariance and zero crosscorrelation with $\mathbf{w_k}$
- $\mathbf{Q_k}$ is the noise covariance matrix associated to the Kalman linear model

In a classical Kalman filter the state estimate is given by:

$$\hat{\mathbf{x}}_\mathbf{k} = \hat{\mathbf{x}}_\mathbf{k}^- + \mathbf{K_k}(\mathbf{z_k} - \mathbf{H_k}\hat{\mathbf{x}}_\mathbf{k}^-)$$

where $\mathbf{K_k}$ is the Kalman Gain computed as:

$$\mathbf{K_k} = \mathbf{P_k^-}\mathbf{H_k^T}(\mathbf{H_k}\mathbf{P_k^-}\mathbf{H_k^T} + \mathbf{R_k})^{-1}$$

The error covariance update is:

$$\mathbf{P_k} = (\mathbf{I} - \mathbf{K_k}\mathbf{H_k})\mathbf{P_k^-}$$

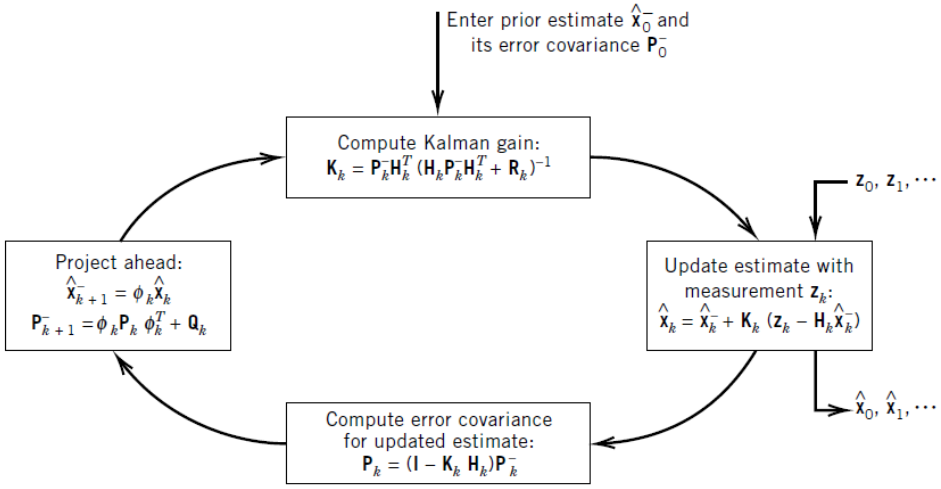

Fig. 6.5 Kalman filter loop. *From [15]*

The state estimate prediction is given by:

$$\hat{\mathbf{x}}_{\mathbf{k+1}}^{-} = \phi_{\mathbf{k}} \hat{\mathbf{x}}_{\mathbf{k}}$$

and the error covariance extrapolation is computed as:

$$\mathbf{P}_{\mathbf{k+1}}^{-} = \phi_{\mathbf{k}} \mathbf{P}_{\mathbf{k}} \phi_{\mathbf{k}}^{\mathbf{T}} + \mathbf{Q}_{\mathbf{k}}$$

With this set of equations we are able to compute accurate PVT solutions by using the received signals, the dynamics of the receiver and its previous known solution. In Fig. 6.5 the illustration of the loop of the Kalman is presented.

6.3.1 Case $N_T - N_s \geq 4$

The Kalman filter under these conditions uses only non-spoofed signals, as defined by the detection and exclusion stage in (6.10). A Kalman filter with eight states is used, considering four states for the position and timing and other four states for

their respective derivatives. The error state vector is:

$$\mathbf{e}_k = \begin{bmatrix} \delta\mathbf{x} \\ \delta t \\ \delta\dot{\mathbf{x}} \\ \delta\dot{t} \end{bmatrix}$$

where $\delta\mathbf{x}$, is the vector related to the position coordinates x, y and z and $\delta\dot{\mathbf{x}}$ is the vector related to the velocity coordinates. This architecture for navigation solution computation is used as soon as the receiver is turned on and it is maintained while $N_T - N_s \geq 4$.

The measurement vector at time t_k is then obtained as:

$$\mathbf{z}_k = \begin{bmatrix} \mathbf{r} \\ \dot{\mathbf{r}} \end{bmatrix}$$

where $\mathbf{r}$ contains the code-based pseudoranges measurements and $\dot{\mathbf{r}}$ is computed using the Doppler measurements obtained from the Phase Lock Loop (PLL). Finally, the matrix describing the connection between the measurements and the state vector at time t_k is given by:

$$\mathbf{H}_k = \begin{bmatrix} \tilde{\mathbf{r}}_{x,y,z} & 1 & 0 & 0 \\ 0 & 0 & \tilde{\mathbf{r}}_{x,y,z} & 1 \end{bmatrix}$$

where $\tilde{\mathbf{r}}_{x,y,z}$ is the unit vector from the user to the satellite, computed using the satellite position and the user's previous positions. This structure of the Kalman filter is the one generally used to obtain PVT solution.

6.3.2 Case $N_T - N_s < 4$

Under these conditions, it is not possible to obtain a solution using only non-spoofed pseudorange measurements. In this case we switch to a Kalman filter version which uses all, N_T, available Doppler measurements. This is because, generally, it is more difficult for the spoofer to enforce a fully controlled PVT computation, by modifying the Doppler measurements, especially if the internal architecture of the receiver is unknown to the attacker.

This version of Kalman filter uses also eight states but provides a PVT solution using only Doppler measurements obtained from the PLL [79, 78]. To compute the solution, the measurement vector $\mathbf{z}_k$ and its connection matrix $\mathbf{H}_k$, are modified to:

$$\mathbf{z}_k = \begin{bmatrix} \mathbf{0} \\ \dot{\mathbf{r}} \end{bmatrix}$$

and

$$\mathbf{H}_k = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \tilde{\mathbf{r}}_{x,y,z} & 1 \end{bmatrix}$$

With these changes, the Kalman filter ignores the spoofed pseudorange measurements and relies only on the phase measurements and previous positions.

6.4 Delay estimation and time jump

In this Section we explain the process of the delay estimation, using the LASSO decomposition, and the modification of the delay being tracked by the DLL, in order to unlock the receiver from the spoofer's control and mitigate its effects.

6.4.1 Delay estimation method

As introduced in Section 6.2, for the delay estimation between LOS and spoofing signal, we use the correlation decomposition performed by the LASSO algorithm. If, during the detection phase, we identify the presence of additional signals in the correlation function, then the distance between the non-zero taps of the filter are representative of the relative delay between them.

This estimated delay has a resolution given by the number of taps of the filter ($\mathbf{w} \in \mathbb{R}^{M\times 1}$) and by the delay between the first and the last correlator T_{range}, e.g. $T_{range} = 2T_{chip} - (-2T_{chip}) = 4T_{chip}$. Using a greater number of taps, maintaining a fixed T_{range}, provides higher resolution in time. So in order to chose the values for T_{range} and M, some considerations need to be done.

Ideal correlation functions are contained in the matrix $\mathbf{U} \in \mathbb{R}^{N\times M}$ whose columns contain shifted versions of the ideal correlation function, with a single point shift

between two adjacent columns. The choice of M with respect to T_{range}, affects the number of possible delayed replicas, reflected in the number of columns of $\mathbf{U}$ and the computational load. Hence, the choice of the resolution needs to meet a trade-off between computational load and the range of the multicorrelator T_{range}. For the experiments presented in this Chapter, we selected $M = 21$ and $T_{range} = 4T_{chip}$ in order to limit the computational load and to have a range wide enough to observe external signals that are well separated from the LOS one.

Once M and T_{range} are decided, we need a technique to estimate the delay $\hat{\tau}$. The intuitive approach would be to observe directly the vector $\hat{\mathbf{w}}$ and count the number of taps between the two coefficients different from zero. This approach is possible if we have only 1 (LOS case) or 2 (LOS + spoofer) coefficients different than zero. Unfortunately this is generally not true because of three factors:

- the presence of noise that could create small additional coefficients (Fig. 6.6)
- the constraint ε in the LASSO, that allows to have more than 2 taps for the approximation when $\varepsilon > 2$ (Fig. 6.7)
- the resolution. Since the true delay τ is not generally an integer multiple of the distance between two adjacent taps, the LASSO will use the combination of two adjacent taps when the LOS or spoofing correlation function peak is between two taps.

We then need a technique that takes into account these factors in order to have a better delay estimation and we called it *Barycenter Delay Estimation* (BDE). The idea is to first remove possible taps with low values, caused by noise, through hard-thresholding. Afterwards, if there are two remaining non-zero coefficients, count the number of taps between them and multiply it by the resolution T_{res}. If the remaining coefficients are more than two, we observe $\mathbf{w}$ and compute the barycenters between adjacent taps. Finally, the delay will be the distance between the two largest barycenters.

Using GPS L1 C/A signal, the correlation functions are all triangles with the same base b and the heights are related to the amplitudes of the coefficients of $\mathbf{w}$. The barycenter will then be computed as:

$$t_{bar} = \frac{iA_i + (i+1)A_{i+1}}{A_i + A_{i+1}} \cdot T_{res}$$

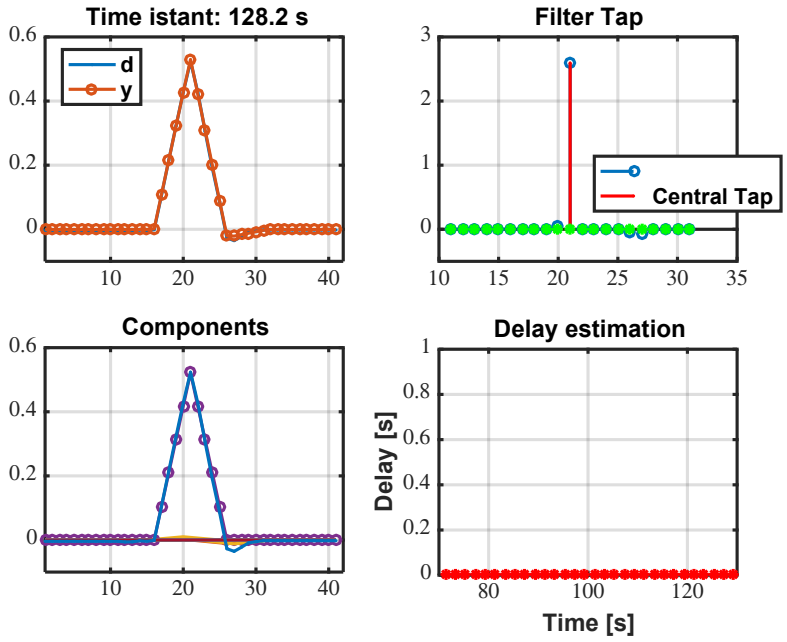

Fig. 6.6 Correlation function decomposition and delay estimation example for a clean scenario with only the true signal and noise. The measured correlation is clean, with only the central tap (0-delayed replica) different from zero. This means that only one signal component is present in signal correlation.

where $A_i = bw_i$ is the area of the triangle associated to the w_i coefficient. Finally, the delay difference $\hat{\tau}$ between the spoofing signal and the LOS, is obtained as:

$$\hat{\tau} = t^1_{bar} - t^2_{bar} \tag{6.11}$$

where t^1_{bar} and t^2_{bar} are the two largest barycenters obtained using (6.4.1). An example of the state of the different elements are shown in Figs. 6.6 and 6.7.

In Fig. 6.8, we show an example of delay estimation vs. time for three satellite signals. We can observe that at the time 110 s, the spoofer starts the push-off phase, and the estimated delay starts growing. After 100∼150 seconds, the estimation stabilizes around a final value for each channel. At the beginning of the attack, from 100 to 150 s, when the spoofing signal is aligned with the satellite signal, the estimation is more difficult because of the small distance between the two peaks. This figure shows that the technique has difficulties in detecting additional signals that are perfectly aligned to the LOS signal. If the signals are aligned in delay and

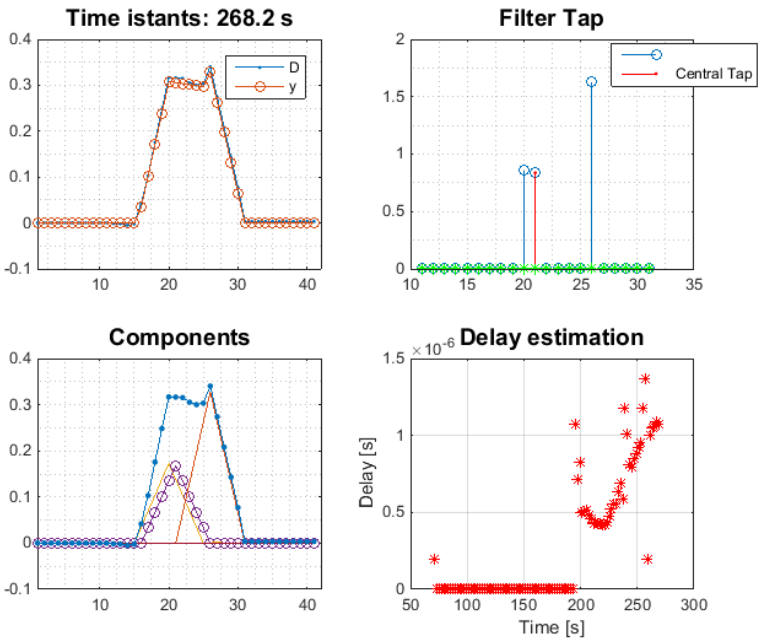

Fig. 6.7 Correlation function decomposition and delay estimation example for spoofed scenario. The distortion in correlation domain is visible also in the number of taps different from zero. Therefore, it is possible to estimate the relative delay between the authentic and the spoofing signal

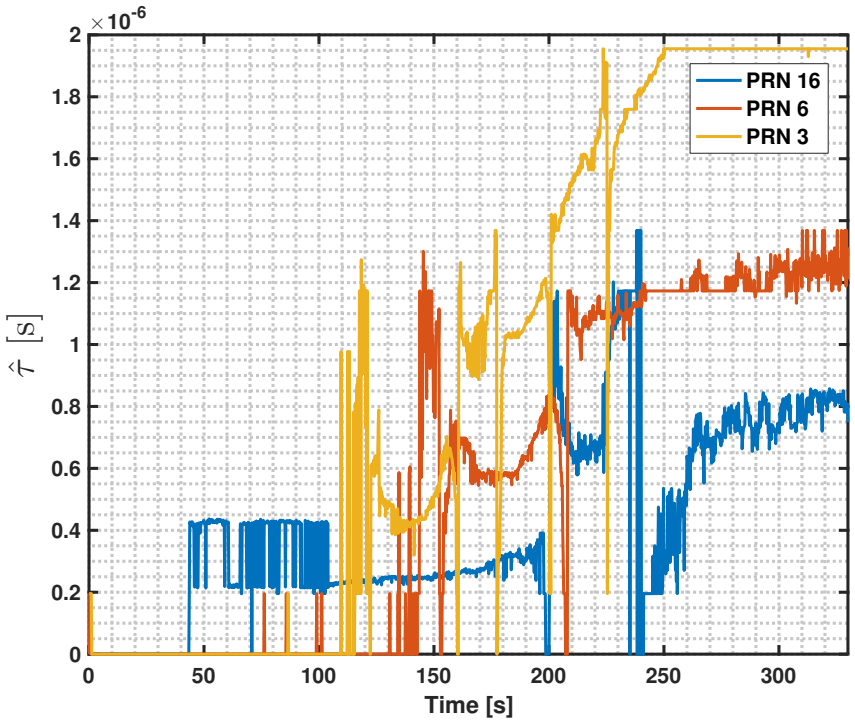

Fig. 6.8 Temporal evolution of delay estimation. Example for 3 different satellites. Not until 100 seconds the spoofer starts the push-off phase of the attack, separating the two peaks from each other. Stable estimation of the delay is obtained around 200 seconds into the test for PRN 6 and 3 and around 250 seconds for PRN 16

phase, no effects will be observed in the PVT solution. On the other hand if the signals are aligned in delay, but the spoofing is in counterphase w.r.t. the LOS signal, the spoofer could perform a navigation data bit attack [35].

After estimating the delay difference between the spoofer signal and the LOS, the algorithm is ready to modify the delay used by the DLL in order to ignore the spoofing signal.

6.4.2 Pre-Jump Checks

In general, during spoofing attacks, the estimated delay $\hat{\tau}$ is a variable parameter, that can be modified by the spoofer, once it takes control of the receiver. Nevertheless, given the nature of a spoofing attack, in order to adequately manipulate the target's position and to avoid that the receiver loses the tracking lock, a spoofer will generally alter the delay information slowly, on the order of 20 ns/s. Therefore, for adequately

small time windows, $\hat{\tau}$ can be considered constant and its variance can be calculated accordingly. We can use the absolute value of $\hat{\tau}$ and its variance, in order to decide if the estimations are correct and the authentic peak is being observed. If these conditions are met, the time jump can be performed during that time instant.

Two basic checks are done in order to identify if the current time is suitable for jumping or not. First we check that the estimated delay is at least 0.5 μs. This heuristic threshold is used to give the possibility for the two signals to be sufficiently apart and for the two peaks to be clearly visible in the correlation function. In case of an unsuccessful jump, an after-jump check is used, as it will be shown in Section 6.4.3.

The second check is observing the variance of the delay estimation within a time window. If the variance during one second of the delay estimation is lower than a predefined threshold γ_σ, the estimation is stable and the channel is ready to jump. We can define the checks as:

$$\hat{\tau} \geq 0.5\mu s \text{ and } \sigma^2_{\hat{\tau}} < \gamma_\sigma \tag{6.12}$$

These checks are done on each channel individually. Finally, we perform the jump on all satellites at the same time, because in case of an attack that leaves at least four non-spoofed satellites, the receiver is able to continue the operation. In Fig. 6.9 we observe an example time instant where the variance of the delay estimation of each channel is below γ_σ. Observing Fig. 6.8 in that time instant, we see that the delay estimations of the satellite signals are greater than 0.5 μs.

Once the checks in (6.12) are positive, the absolute delay of the DLL is modified by $\hat{\tau}$, in order to unlock the signal from the spoofer, and lock it into the authentic signal.

6.4.3 Post-Jump Checks

After the jump has been performed, we need to confirm that the DLL is tracking the satellite signal. We observe the value of $\hat{\tau}$ and we verify the behavior of the tracking loop by observing the correlators outputs. It is possible to define three possible outcomes for each satellite and they are shown in Fig. 6.10. These outcome are:

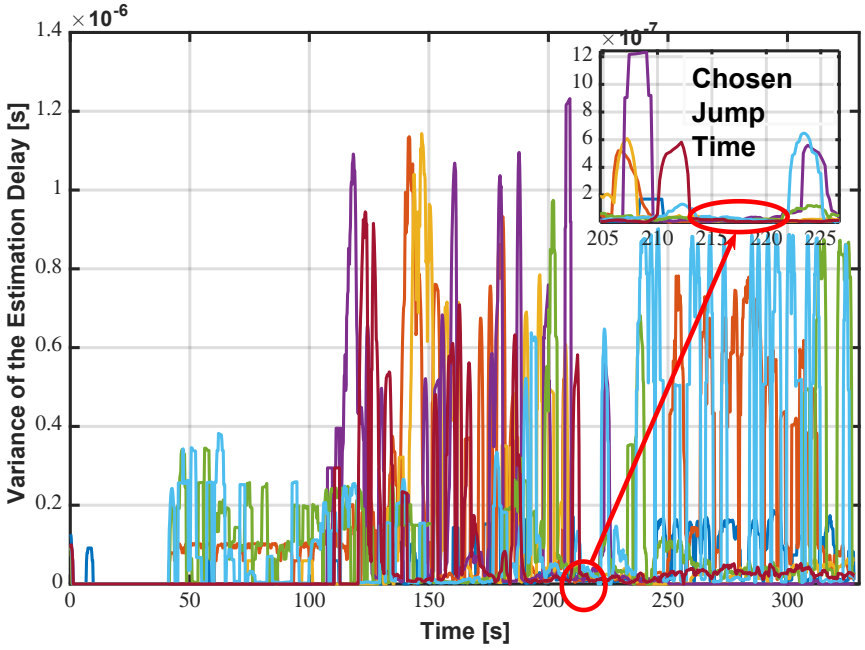

Fig. 6.9 Variance of the estimated delay for all visible channels. The red circle highlights the time instants chosen to jump because they have the lowest estimated variance

- *Successful jump* (red line of Fig. 6.10). In this case, the jump successfully unlocks the signal from the spoofer and ends up locked into the authentic signal. A change in the sign of $\hat{\tau}$ is observed, meaning that the DLL successfully jumped from one peak to the other. Also, the correlator level will be above zero, indicating that a signal is being tracked.

- *Unsuccessful jump* (blue line of Fig. 6.10). In this case, the jump unlocks the DLL for a limited time, but it returns to lock itself to the spoofing signal. In this scenario, $\hat{\tau}$ does not change signs and maintains a similar value before and after the jump. Also the correlator levels indicate that a signal is being tracked.

- *Loss of lock* (green line of Fig. 6.10). In this case, the jump modifies the DLL to a point where there is no signal present and it is not able to lock itself to any peak. In this scenario the correlator levels are very low indicating that no signal is being tracked and a value for $\hat{\tau}$ cannot be obtained accurately. When this occur, the channel should go back to the acquisition stage and restart the process.

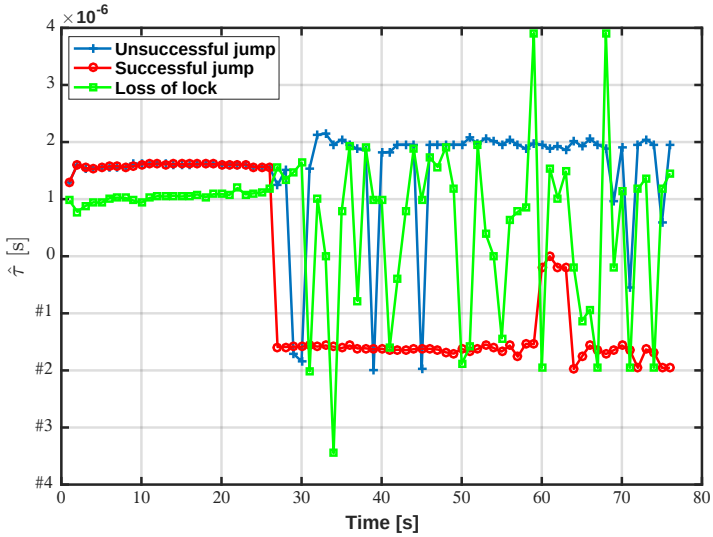

Fig. 6.10 Delay Estimation of three possible outcomes after the Jump (at second 26): loss of lock (green), successful jump (red) and unsuccessful jump (blue)

In Fig. 6.11 we show a graphical explanation of the three possible outcomes of the jump procedure. We observe the three possible scenarios, and what is the outcome on the DLL once the filter has stabilized. We observe that the loss of lock scenario could be produced by the wrong delay estimation $\hat{\tau}$.

After performing the jump on the DLL, and verifying the tracking of each channel, the TJ algorithm declares as un-spoofed, the channels where the jump was performed correctly, modifying N_s, and possibly the Kalman filter used for navigation calculation. Channels where the jump is *unsuccessful* or *lost the lock*, are monitored with the SQI, and another jump attempt should be performed for spoofed satellites, if the conditions described in Section 6.4.2 are met.

6.5 Results for time jumper algorithm

As stated previously, the TJ algorithm aims at detecting spoofing attacks and providing continuous use of GNSS signals in the receiver. It also has the scope of mitigating the effects of these attacks by means of the delay jump.

In order to asses the TJ algorithm performance we use the TEXBAT datasets [38], describe in Appendix A.1. From the different datasets available in the test bed

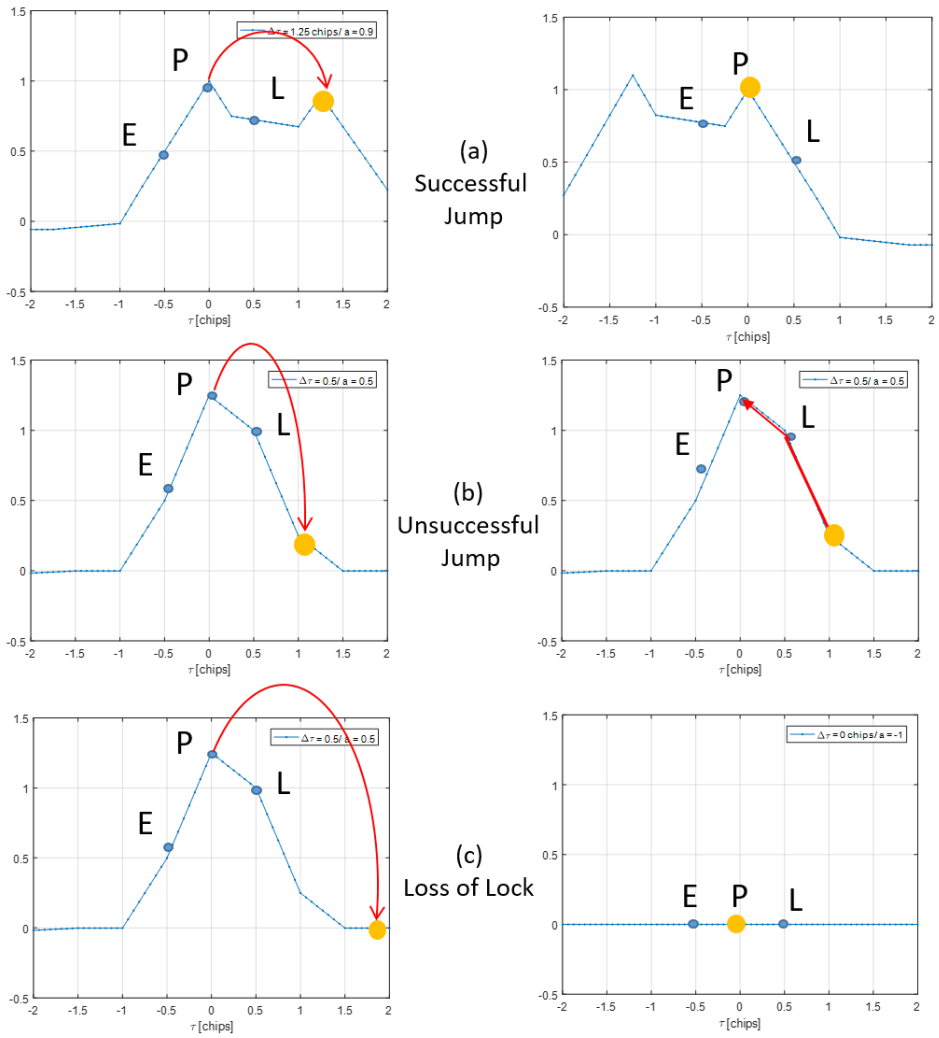

Fig. 6.11 Graphical explanation of the three possible outcomes of the jump. In (a), we can observe how a successful jump is able to jump from one peak to the other, thus locking itself to the satellite signal. In (b), an unsuccessful jump is shown, where the DLL locks back to the previous peak. In (c) a loss of lock scenario is shown, where the delay estimation is not accurate, making the DLL land where no signal is present.

we focus on the scenario ds6, which consists on a dynamic matched-power position push. We believe that this is a good example of a typical vehicular application and serves the purpose of demonstrating the feasibility of the TJ algorithm. We additionally present results for the static scenario ds4. With these results we show the functionality and capability of the TJ algorithm, in both dynamic and static cases.

For each scenario, four cases were studied in order to highlight the overall working procedure:

- Case 1 is the clean solution obtained using a generic GNSS software receiver and where no spoofing attack was present. It serves as a reference solution for the other cases.
- Case 2 is the solution obtained with the generic software receiver for the spoofing attack scenario. This provides the reference for the situation where no anti-spoofing defense is present in the receiver.
- Case 3 is a solution using only the jumping procedure together with a classical PVT computation based on a Kalman filter using code and Doppler measurements. It serves to demonstrate the effects of the jump and the advantages of using the double choice Kalman filters of case 4.
- Case 4 is the solution using the TJ algorithm, i.e. jumping procedure and the double choice Kalman filter.

Fig. 6.12 shows a comparison of the errors of x, y and z compared to case 1. For the case 2, shown in blue, we observe that the solution is driven away from the reference and large errors are introduced mainly in the z axis. For the case 3, in orange, the effects that the jump has in the overall solution are clearly visible, at time 270 seconds. We observe that after the jump, the solution comes back to the real path and the maximum errors are reduced. Finally, we can observe the performances of the complete TJ algorithm, shown as case 4, in yellow. The solution follows closely the real path and it is never controlled by the spoofer.

In Fig. 6.13 the 2D tracks are presented for the four cases. It is possible to observe that by using the TJ algorithm, we are able to provide a continuity of unspoofed solutions, contrary to what is observed for case 3, in orange, where the error increases considerably before the jump is feasible. For TEXBAT scenario ds6, from the six

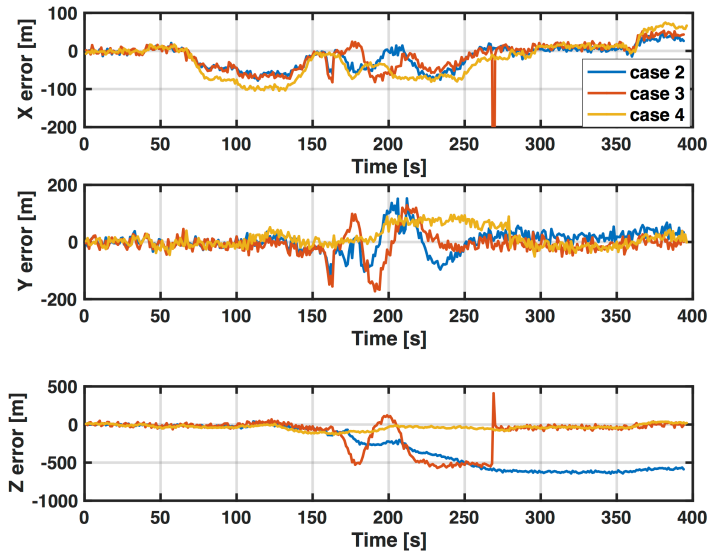

Fig. 6.12 x, y, z results for scenario ds6 in the different cases. The blue line is the position error in case two. In orange the third case with only the Jump and in yellow the error when using the TJ algorithm.

satellites tracked before the jump, five satellites performed a successful jump and one lost the lock, so it was excluded from the solution after 270 seconds of the test.

For the static case, scenario ds4, results are presented in Figs. 6.14 and 6.15. We observe how in this scenario the Kalman filter is not working as well as in the dynamic scenario. The low dynamics of the scenario make the Doppler measurements less informative for PVT usage.

In the static case, from the six satellites tracked before the jump, four of them did a successful jump, one did an unsuccessful jump and one lost the lock, so the final PVT solution after the jump is performed with four satellites. Nevertheless we observe how the errors decrease considerably when using TJ algorithm and the solution is reliable throughout the whole test.

In Table 6.1 we can observe how the 3D rms error is reduced considerably between the different cases. If we compare cases 2 and 3, we observe how the jumping procedure alone, reduce the mean error in more than 55 % for the ds6 scenario and 40 % for scenario ds4. Comparing case 2 and 4, the TJ algorithm improves the mean and standard deviation of 3D rms error in more than 80 % for both scenarios. We observe how in case 2 we have the maximum error, but this is

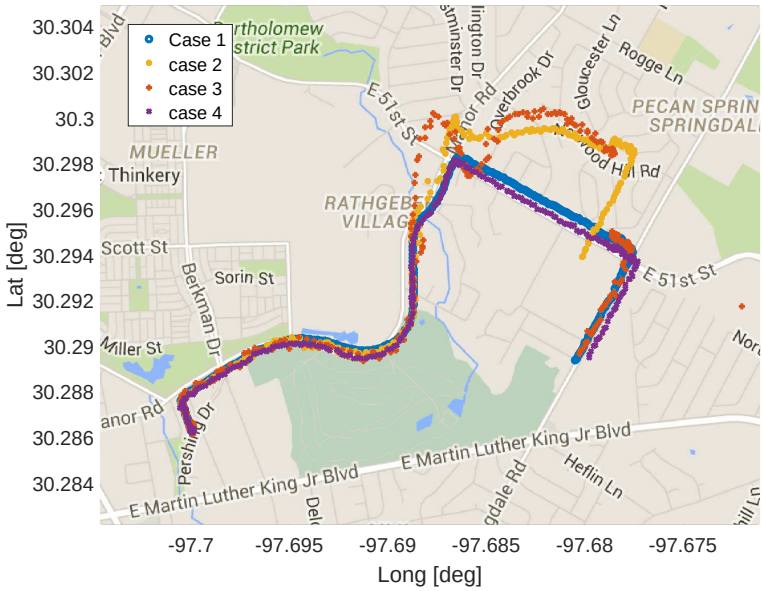

Fig. 6.13 The dynamic track of ds6 for the different cases over map. The blue line is the real path of case 1, the orange line is the spoofed track of case 2. In yellow is depicted the path of case 3 and in purple the case 4 path, using the TJ algorithm

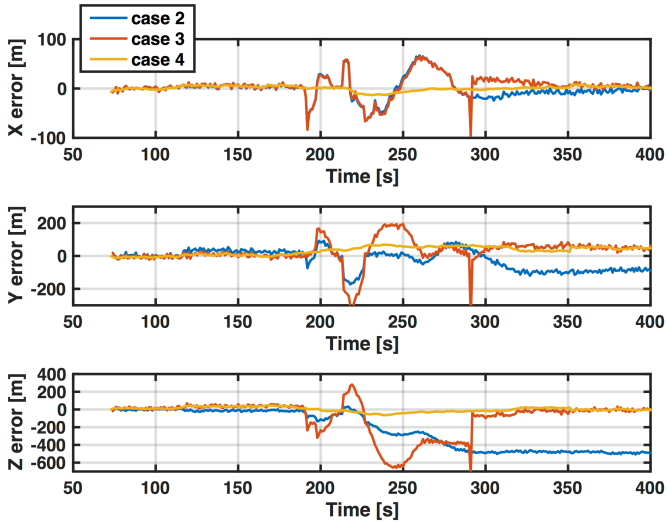

Fig. 6.14 x, y, z results for scenario ds4 for the different cases. The blue line is the position error in case two. In orange is the error for case 3 with only the Jump. In yellow is the error of case 4, using the TJ algorithm

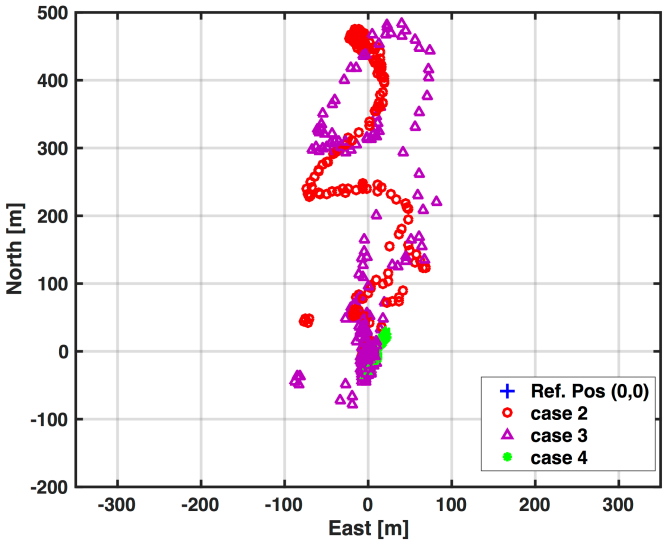

Fig. 6.15 The track of ds4 for the different cases. The blue point is reference static position, the red ones indicates the spoofed track of case 2, in purple is depicted the path for case 3 and in green the path of case 4, using the TJ algorithm

Table 6.1 Mean, standard deviation and maximum 3D rms error in meters, for each of the 3 cases and for scenarios ds6 and ds4

	ds6			ds4		
3D rms error	mean	std	max	mean	std	max
case 2	413	218	645	290	195	512
case 3	181	193	747	170	186	846
case 4	82	30	134	49	16	94

generated from the spikes generated from the jump, as can be observed in Figs. 6.14 and 6.12.

Conclusions

In this Chapter we presented a novel anti-spoofing algorithm, that can be used for detection and mitigation of spoofing attacks. The TJ algorithm make use of different signal processing techniques and combine them in order to protect the receiver against the effect of spoofing. It was also tested against two of the TEXBAT datasets and promising results were obtained for the mitigation of the effects. The Kalman filter, using only Doppler measurements for navigation solution computation, represents an upgrade with respect to the regular Least square solution and provides another level of protection against the attacks.

We can observe that in order to detect and mitigate the spoofing effects, the TJ algorithm uses complex algorithms for signal decomposition, quality monitoring and PVT computation. Combining all of these results in a complex technique with a heavy computational load that would require significant software optimization to be implemented in a real receiver. Nevertheless, the mitigation capabilities of the techniques were demonstrated and no external hardware is required for its implementation, making it a desirable technique for receivers that need to have stand alone mitigation capabilities and do not have constraints in the computational requirements.

Chapter 7

Feedback tracking architecture for spoofing mitigation

In this Chapter, we present an anti-spoofing detection and mitigation technique that uses a dedicated tracking architecture in order to mitigate the effects of the spoofing signal. This Chapter revisits and improves a tracking technique known as Extended Coupled Amplitude Delay Lock Loop (ECADLL), originally presented in [18, 17]. The goal of the work is to tailor the architecture to spoofing detection, making it also able to mitigate the effects created by spoofing attacks. Moreover, we define an effective algorithm for spoofer detection with the capability to distinguish between multipath and a spoofing attack. The ECADLL was tested against realistic spoofing scenarios, and the performance was compared against the SQM as presented in Chapter 4. With the introduction of a *ratio test*, the detection latency was reduced, and the computational load of the algorithm was decreased.

This Chapter is based on the work presented in [53].

7.1 The ECADLL concept

The working principle of the ECADLL is based on distinguishing, and tracking separately, the satellite signal and the impairment signal, i.e., spoofer or multipath. It uses an architecture made of multiple DLLs and a feedback loop to remove the extra components from the incoming signal and track each signal individually. The

incoming signal at the baseband level for a single satellite, denoted as $s(t)$ and under impairment presence denoted as $m(t)$, can be written as:

$$s(t) = a_0 D(t-\theta_0) c_f(t-\tau_0) e^{j(2\pi(f_{IF}+f_D)t+\theta_0)} + m(t) \tag{7.1}$$

where,

$$m(t) = D(t-\theta_0-\theta_n) \sum_{n=1}^{M} a_n c_f(t-\tau_0-\tau_n) e^{j(2\pi(f_{IF}+f_D)t+\theta_0+\theta_n)} + n_f(t) \tag{7.2}$$

and:

- $c_f(t)$ is the spreading code of the signal
- τ_0, a_0 and θ_0 are the satellite signal code delay, amplitude and carrier phase, respectively
- τ_n, a_n and θ_n denote the code delay, amplitude and carrier phase of the n-th impairment ray with respect to the LOS
- f_{IF} is the intermediate frequency of the front-end
- f_D is the Doppler shift of the signal carrier
- $D(t)$ is the navigation data information
- $n_f(t)$ is the Gaussian noise

In (7.1) and (7.2), we observe the overall signal that is used by the receiver after the RF front-end filtering and the IF down-conversion. The term (7.1) shows the signal coming from the satellite, while (7.2) contains all possible impairment rays along with a model of the noise of the signal.

Throughout this Chapter and in general for spoofing detection, $M = 1$ will be assumed, due to the fact that only one powerful ray is assumed to be present during a spoofing attack. As have been mentioned previously, the spoofing signal has the same structure as the satellite signal and this can be observed in (7.2). The ECADLL uses this intrinsic similarity to track each ray individually, using a special tracking structure referred to as the unit. Each unit consists of a unique DLL, plus a couple of Amplitude Lock Loops (ALL), detailed in [18]. The objective of each unit is

to estimate the delay τ_n, the amplitude a_n and the phase θ_n of the different n rays, inside each channel.

The input signal of the i-th unit will be:

$$s_i(t) = s(t) - \sum_{n \neq i}^{N} \hat{a}_n \cdot c(t - \hat{\tau}_n) e^{j(\hat{\theta}_n)} \tag{7.3}$$

where $\hat{\tau}_n, \hat{a}_n$ and $\hat{\theta}_n$ are the estimated values obtained from the tracking of unit$n$. If the process is done correctly, for $N = 1$, we would obtain:

$$s_0(t) = a_0 c_f(t - \tau_0) e^{j(\theta_0)} + n_f(t) \tag{7.4}$$

$$s_1(t) = a_1 c_f(t - \tau_0 - \tau_1) e^{j(\theta_0 + \theta_1)} + n_f(t) \tag{7.5}$$

From (7.5) and Fig. 7.1, the working principle of the ECADLL can be understood. Using a feedback loop, it subtracts from the overall received signal the sum of the estimated signals in other units. In this way, the structure is able to separate the impairment component from the satellite signal and track each one on a different unit.

As can be observed in Fig. 7.1, the total structure for one GNSS channel has a single PLL, plus one unit for each additional signal that has to be tracked. In the case considered in this work, the structure will have two units, one for the LOS signal coming from the satellite and the other to track the spoofing signal.

Inside each unit, a DLL using a narrow correlator is used to estimate $\hat{\tau}_n$ and uses the normalized dot-product discriminator:

$$d\tau = \frac{I_D \cdot I_P + Q_D \cdot Q_P}{S_P} \tag{7.6}$$

where I_P and Q_P are the prompt correlations of the in-phase and quadrature signals, respectively. $S_P = I_P^2 + Q_P^2$ and I_D and Q_D are the early-minus-late of the I and Q channels.

Inside the ALL, the amplitude $\hat{a}_n$ and phase $\hat{\theta}_n$ are estimated as:

$$\hat{a}_n = \sqrt{\hat{a}_{n,i}^2 + \hat{a}_{n,q}^2} \qquad \hat{\theta}_n = tan^{-1}\left(\frac{\hat{a}_{n,q}}{\hat{a}_{n,i}}\right) \tag{7.7}$$

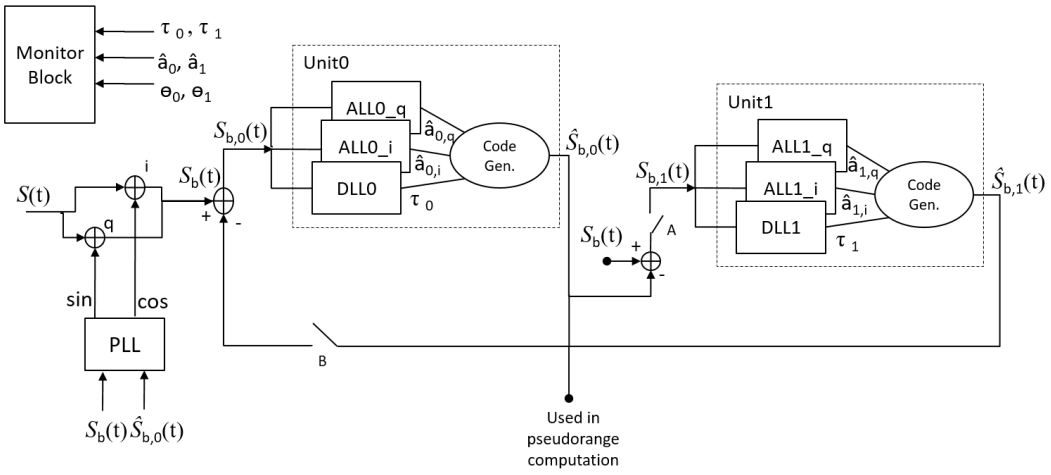

Fig. 7.1 Extended Coupled Amplitude Delay Lock Loop (ECADLL) architecture for a single GNSS channel and configured for spoofing detection.

where $\hat{\theta}_n$ is the estimation of θ_0 for $n = 0$ and $\theta_0 + \theta_n$ for the others. Amplitude values, $\hat{a}_{n,i}$ and $\hat{a}_{n,q}$, are estimated using the in-phase and quadrature results of the prompt correlation I_P and Q_P, as:

$$\hat{a}_{n,i} = \frac{I_P}{\lambda} \tag{7.8}$$

$$\hat{a}_{n,q} = \frac{Q_P}{\lambda} \tag{7.9}$$

where λ is a damping factor, slightly smaller than one, used to adjust the estimation accuracy.

The PLL structure in Fig. 7.1 does not use the same correlation output as other units. It uses the correlation values between the total incoming signal ($s(t)$) and the Unit0 local code ($c(t - \tau_0)$). Thus, in the presence of impairments, the local carrier phase will not be completely aligned with the carrier phase of the satellite signal. Nonetheless, the carrier phase error will be estimated by the couple of ALLs inside each unit. Recovering the relative phase shift for any single signal component is necessary, in order to produce the correct replica and add it to the feedback signal.

As an example, in Fig. 7.2, the basic working procedure of the ECADLL algorithm is shown when the spoofing and satellite signals are aligned in phase. On the left-hand side of the picture, we show an estimated correlation function of a generic received signal affected by spoofing. On the right-hand side, we observe the results for the first iteration and for the iterations after the architecture reaches

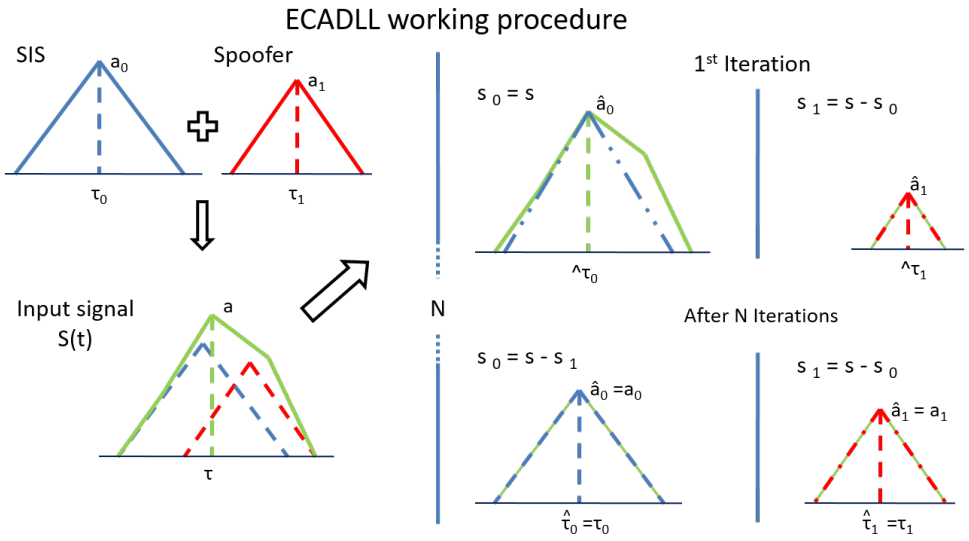

Fig. 7.2 ECADLL basic working procedure when the satellite signal and the spoofing signal are aligned in phase. On the left, the incoming signal is shown, and on the right, the solution after N iterations. Only the in-phase channel is shown in this image for simplicity.

the steady state. At the beginning of the tracking procedure, the architecture starts tracking the residual of the operation $s(t) - \hat{s}_0$ inside Unit1. At the following iteration, the feedback Switch B is closed, and after a transition time, the two units' estimations will be locked at values very close to the originals, such that $\hat{s}_0 \approx s_0$ and $\hat{s}_1 \approx s_1$.

A monitoring block, shown in Fig. 7.3, receives as inputs the estimations of $\hat{\tau}_n$, $\hat{a}_n$ and $\hat{\theta}_n$, for all of the units, and it is in charge of turning on and off each of them. We worked under the assumption that a spoofing attack will happen after some time that the receiver is turned on, so the monitoring block is in charge of deciding the moment when each unit is to be introduced by closing Switch A.

If the spoofer is not present, Unit1 will be tracking noise, so its amplitude $\hat{a}_1$ will be small, and its delay $\hat{\tau}_1$ will wander randomly. Once the spoofer signal appears on the satellite signal correlation space, Unit1 will quickly track the interference signal, thus increasing its value of amplitude and locking the DLL around a value of τ_1. At this moment, the monitoring block will close Switch B, starting the feedback loop and obtaining a better estimation of the parameters.

The thresholds used by the monitoring block are defined based on the product of front-end bandwidth (B) and the time of chip (T_c). The product $B \cdot T_c$ affects the shape of the correlation function, rounding the peak of the correlation when the

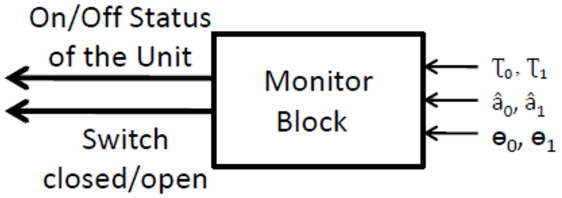

Fig. 7.3 Monitoring block for ECADLL. It receives as inputs the estimations of delay, amplitude and phase, and it controls the powering and insertion of units inside the loop.

product is small. This effect causes a degradation on the detection capabilities, for impairments that are close to the peak.

After presenting the basic working principle of the ECADLL architecture, following Sections present novel contributions of this thesis. In Section 7.2 we discuss the spoofing detection algorithm, which uses as inputs the estimations of each unit, i.e., $\hat{\tau}_n$, $\hat{a}_n$ and $\hat{\theta}_n$, and takes the decision of whether an impairment is present in the incoming signal or not.

7.2 Use of ECADLL information for spoofing detection

The ECADLL architecture was originally proposed as a multipath mitigation technique. It is able to eliminate the tracking errors under multipath scenarios for signals having a delay larger than ≈50 ns relative to the satellite signal [18]. The goal of this Section is to study and propose a reliable detection technique, able to recognize the spoofer's presence. In Fig. 7.4, a basic block diagram is presented, where $D_i(t_k)$ is the decision variable for the i-th channel at time t_k. Its values are defined as:

$$D_i(t_k) = \begin{cases} 2 & \text{Spoofer} \\ 1 & \text{Impairment} \\ 0 & \text{No impairment} \end{cases} \tag{7.10}$$

It is important to notice that the information provided by the ECADLL architecture is soft information on the estimation of the delay, amplitude and phase of the different signals ($\hat{a}_0 \ldots \hat{a}_n, \hat{\tau}_0 \ldots \hat{\tau}_n, \hat{\theta}_0 \ldots \hat{\theta}_n$). This information is combined to be used as

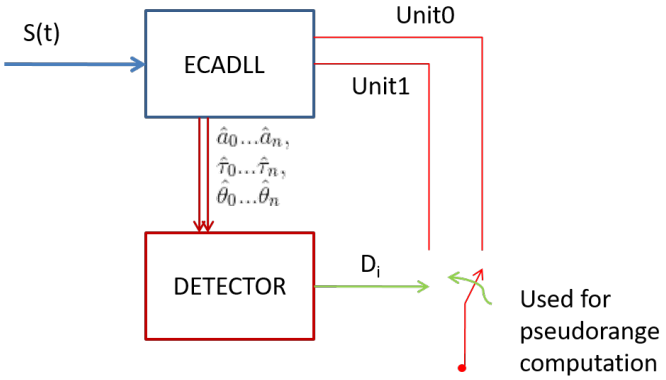

Fig. 7.4 Decision making block diagram.

an effective spoofing detection method and has the potential to help with estimating the real user position under spoofing attacks.

7.2.1 Detecting a generic impairment

The first thing the algorithm focuses on is detecting a generic impairment. By providing this information, the receiver is put into an alert stage, and it does not trust the channels that are flagged as impaired.

In order to detect that an undefined impairment signal is present, either multipath or spoofing, an initial decision can be made using the basic information of Unit1. If the DLL in Unit1 is locked, thus having a low variance of $\hat{\tau}_1$, and the relative amplitude $\hat{a}_1/\hat{a}_0$ is greater than a certain threshold $\hat{a}_{th}$, the channel can be declared as impaired, and the error mitigation for multipath starts working.

$$D_i(t_k) = 1 \text{ if } \sigma_{\hat{\tau}_1} < \sigma_{th} \text{ and } \frac{\hat{a}_1}{\hat{a}_0} > a_{th} \tag{7.11}$$

where $\sigma_{\hat{\tau}_1}$ is the variance of the estimated delay $\hat{\tau}_1$ in a 1-s window. A window of 1-s duration was used, as a tradeoff providing enough data points to have a good estimation of variance $\sigma_{\hat{\tau}_1}$, but keeping the capability to make a decision at a frequently enough rate. Thresholds σ_{th} and a_{th} are defined according to the DLL bandwidth and the expected level of noise. In order to assess the threshold σ_{th}, we observed the relationship between the variance of the DLL when tracking a simulated interference signal and the variance when no additional signal was

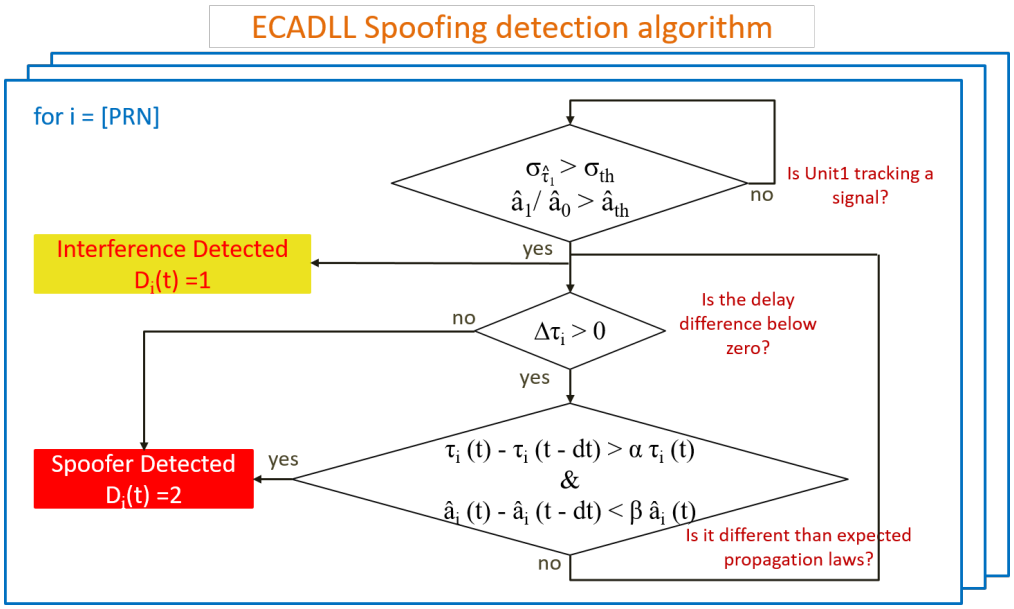

Fig. 7.5 Figure of the proposed ECADLL algorithm. The procedure is repeated for all of the satellites, and it is able to classify the signal tracked by Unit1 as either impairment or spoofing.

present. Using these values, we were able to obtain an appropriate threshold for the specific configuration. For threshold a_{th}, we empirically chose to flag any signal with at least 10% of the Unit0 signal power, since any signal stronger than that is likely to be an interference signal.

In Fig. 7.5, we can observe how the top-most decision is the discrimination between classifying the signal as impairment and no impairment. Summarizing, the channel will be flagged as impairment if the tracking of Unit1 is considered locked to a signal.

7.2.2 Classification of spoofing presence

The generic impairment detection is the first step of the detection algorithm. Once it has been classified as impairment, the goal is to recognize whether or not it can be labeled as a spoofing attack. In order to effectively declare the impairment as a spoofing signal, different aspects need to be taken into account. From now on, the signal with the higher amplitude $(\hat{a}_n)$ will be declared as the spoofing signal and the other one as the real signal, for the sake of the tests to follow.

Detection of negative delays

A simple check that can be performed, in order to decide whether or not a spoofer is present, is to observe if the relative delay is negative, such as:

$$\hat{\delta}\tau = \tau_1 - \tau_0 < 0 \tag{7.12}$$

In this case, a spoofing attack can be declared safely assuming that the Unit0 is tracking a non-zero level satellite signal. Given the physical nature of the multipath, and assuming that the satellite signal has a non-zero power level in the receiver, it is not possible for a multipath reflection to have a delay smaller than the LOS signal, because, by definition, the LOS is the minimum distance between receiver and satellite.

On the other hand, a spoofing signal could perfectly arrive into the receiver before the satellite signals, thus giving away its presence. A spoofer could avoid this by managing its delay correctly, but this simple check is a first hard detection to exclude these cases. Following this first discrimination, the impairment will always be declared as a spoofer when (7.12) is true, and we can observe in Fig. 7.5 how it is implemented as a second level of discrimination.

Detection by using physical laws of propagation

One of the main differences between the spoofing signal and the multipath signal is that the latter is bounded by the physical laws of propagation. Due to these physical limitations, it is not possible for the multipath signal to increase its relative delay $\hat{\tau}_1$, while maintaining the same amplitude $\hat{a}_1$. This is due to the fact that, in order to have a longer delay, the satellite signal needs to travel a longer physical path, and as a result, its power level will decrease. This consideration means that a single multipath signal will always have a smaller amplitude when the delay is larger. These bounds do not apply to spoofing signals. This remark is also valid the other way around, where a closer delay is bounded to have a greater amplitude.

A spoofing attack may be able to replicate the signal propagation behavior of the signal when it attempts to align the spoofing signal with the authentic satellite signal. Nevertheless, when the spoofing signal takes control of the receiver and the push-off phase starts, the satellite signal will be present in the correlation domain, and it

will be bound to the physical laws. The satellite signal, observed in the correlation domain, will have the same amplitude regardless of the relative delay to the spoofer.

This can be defined for time t_k and the i-th channel as:

$$\text{if } \tau_1(t_k) - \tau_1(t_k - dt) > \alpha\tau_1(t_k) \tag{7.13}$$

$$\text{and } a_1(t_k) - a_1(t_k - dt) < \beta a_1(t_k) \tag{7.14}$$

where dt is the time difference between the two measures. Coefficients α and β are smaller than one and are defined using the expected propagation law. When the conditions (7.13) are met, the decision for spoofing attack is made, i.e., $D_i(t_k) = 2$. In our case, we assumed a conservative linear decay, so both α and β are defined as 10%, meaning that a 10% increment in delay will require at least a 10% decay of the amplitude in order to not declare it as a spoofer. Once a channel is declared as being spoofed, it will maintain its status as long as the DLL in Unit1 is locked. It is important to mention that these statements hold as long as the units are tracking continuously the signal between times $t_k - dt$ and t_k. If, for any reason, the Unit1 loses track of the signal, the check is not valid anymore.

In Fig. 7.6, the different regions of the decision are graphically shown. Taking into consideration these behaviors, we can distinguish between generic impairments and spoofing attacks. It is important to notice that these remarks are assuming the spoofer will behave in a specific way, so if the spoofer does not follow these criteria, the receiver will be only able to label it as a generic impairment and not a spoofing attack. Nonetheless, the assumptions presented above are common during spoofing attacks, and in the cases where the spoofer is more harmful, it will likely fall into one of the defined regions.

Additional remarks for spoofing classification

One additional remark that can be made is based on the knowledge of the dynamic behavior of the receiver, e.g., checking for the continuity of the impairment signal over time if the user is in a dynamic environment. In dynamic scenarios, multipath rays will be appearing and disappearing as the physical scenario (objects, buildings,etc.) changes around the receiver. In these cases, having an impairment signal that is persistent for several seconds is highly improbable, so a spoofing attack could be declared.

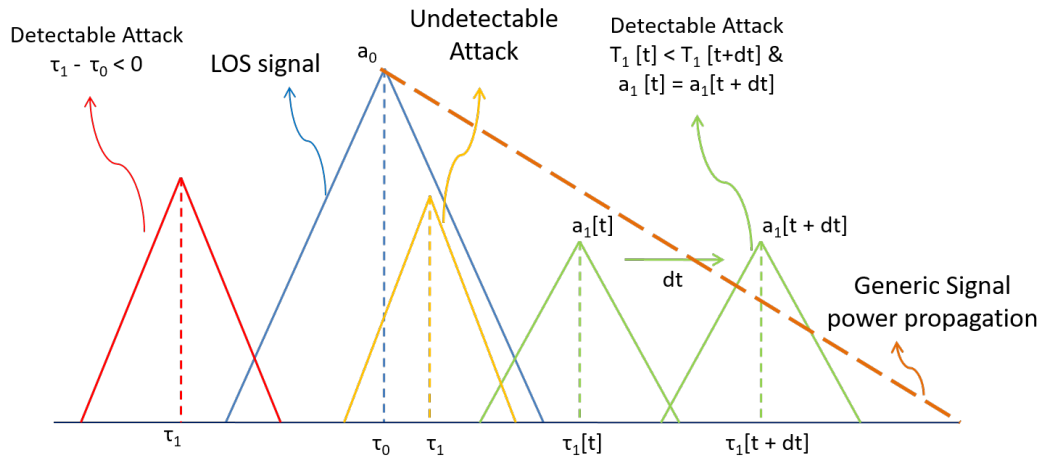

Fig. 7.6 Impairment distinction based on soft observations. The signal in red is detected as a spoofing attack because the delay difference $\tau_1 - \tau_0$ is negative. The green one is detected because the delay increases, but the amplitude is maintained constant, and the yellow one is classified as impairment because there is not enough information to allow discrimination.

On the other hand, in a static scenario where the receiver is fixed, this assumption does not hold anymore, because a reflection from a nearby object can be persistent over time. Nonetheless, in a static case, the receiver can be declared as spoofed once the error in the PVT surpasses a certain value that a multipath signal will unlikely produce, this assumption only holds if the antenna position is georeferenced, such as in permanent GNSS reference stations. In static cases, it is also important to remember that multipath signals can be observed and classified a priori if the receiver is in a known and fix environment, so anything that is not the known signals can be also classified as spoofing. All of these considerations are heavily case dependent, and that is why they were not included in the general classification algorithm, but they could be effectively used to adapt the algorithm to a given scenario.

7.2.3 Improving computational load and detection latency

One drawback of the use of basic ECADLL is that when no additional signal is present, Unit1 of each channel is continuously searching for it over random delays. This behavior creates additional computational burden and may cause a delay in the estimation of the real characteristics of the spoofing/multipath signal. Given that the delay estimation is wandering randomly, it may take a longer time for it to converge to the true delay once the additional signal is present. In order to improve

this situation, in our implementation, we control the turning on/off of the Unit1 by detecting distortions on the correlation function using a *ratio test*.

It has been proven that the SQM using RM works well detecting distortions in the correlation function, as was presented in Chapter 3 and 4. Unfortunately, these techniques are not able to estimate the delay of the spoofing signal or mitigate the errors caused by them in the position and time solution. For ECADLL, using just one RM to detect distortions in the correlation functions and as an additional input to the monitoring block, the estimation and detection latency of the spoofing signal and the computational load can be improved.

In the RM, M can be defined as defined in (3.2). In this case, the monitoring block computes the mean value of M during a 1-s time window, and if it surpasses a pre-defined threshold based on the probability of false alarm, Unit1 is inserted in the feedback loop in order to detect the signal generating the distortion as quickly as possible. A 1-s time window was again chosen as a trade-off to obtain a correct estimation of the mean value and the time needed by the algorithm to make a decision. For spoofing detection, a 1-s time window is still well below the time that a typical spoofing attack will need to gain control of the receiver, which usually extends to several minutes [38].

In Fig. 7.7, we can observe the impact on the detection latency when using the RM. When using RM inside the monitoring block, the spoofing attack is detected quickly after the push-off phase starts, around 120 s. Following this initial detection, the metric has some misdetection, and then, it recovers and declares the spoofing attack for the duration of the test. Furthermore, in Fig. 7.8, we can observe that the delay estimation does not change and is less noisy during the time when the attack starts. Having Unit1 turned off for the first 120 s will greatly help in the computational time for the scenario.

In Section 7.4.3, we provide numerical and analytic results for the improvement of the computational load and the impairment detection latency.

7.3 Experimental setup and baseline results

In order to assess the capabilities of the ECADLL technique under spoofing attack, the TEXBAT dataset was used [38]. The TEXBAT is described in Appendix A.1 and for the results presented hereafter we used all the matched-power scenarios,

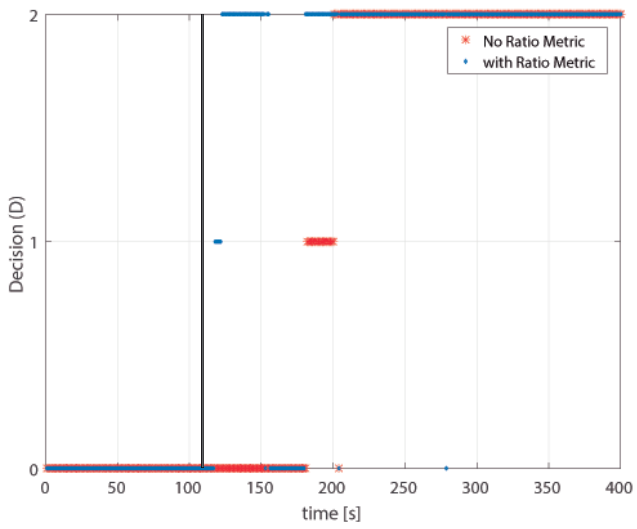

Fig. 7.7 Decision made for a spoofing attack scenario. In red is depicted the decision when no RM is used and in blue when using RMs inside the monitoring block. The start of the spoofing attack is shown in black for reference.

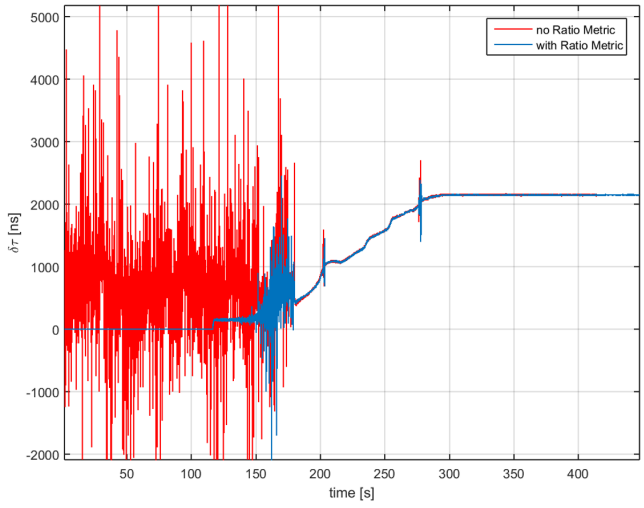

Fig. 7.8 Delay estimation for a spoofing attack scenario. In red is depicted the decision when no RM is used and in blue when using RMs inside the monitoring block. The start of the spoofing attack is shown in black for reference.

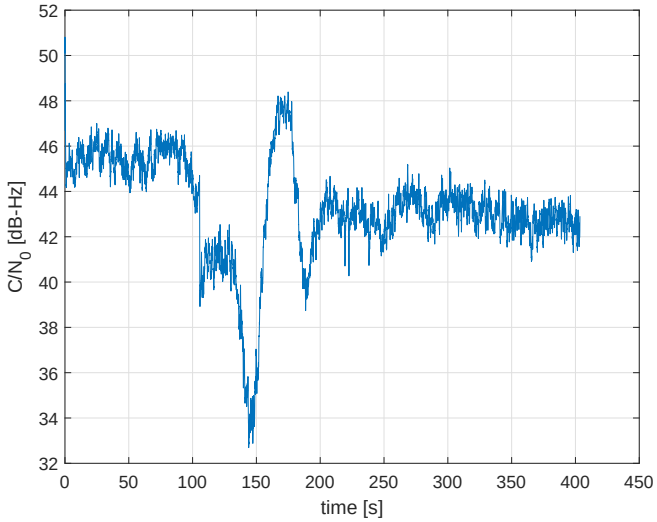

Fig. 7.9 C/N_0 for satellite 6 using TEXBAT ds3.

including the updated ds7. For the reminder of this Chapter, the Static Clean scenario of the TEXBAT will be referred as *clean*.

It is important to notice that the results presented here were obtained by transmitting the TEXBAT dataset by cable connection to a GPS L1 front-end, named SIGE2 [1], and processing the results using the ECADLL software receiver. By doing so, the signal is filtered through a front-end filter, which has a 4-MHz bandwidth and a resolution of two bits. Furthermore, the sampling frequency is 16.3676 MHz, and the data were processed at an intermediate frequency of 4.1314 MHz. This configuration provides a lower resolution than the original TEXBAT files, but gives a more common front-end configuration similar to the ones found on commercial receivers.

Results for TEXBAT ds3 are presented hereafter. Results are presented for one of the channels, but similar behaviors are obtained for the others. In Fig. 7.9, the estimated C/N_0 of PRN 6 is presented. It can be observed that it has a similar trend to the one shown in [38], but with a small loss of power, of a few dBHz, due to the different front-end configuration and re-transmission.

In Fig. 7.10, the delay difference $\delta\tau = \tau_1 - \tau_0$ is plotted. In this figure, the functionality of the ECADLL can be observed: during the first 120 s, the estimated delay is zero, as Unit1 is still turned off. When the asymmetry is revealed, after 120 s, it locks in to the signal that is separated from the spoofing signal, being tracked by

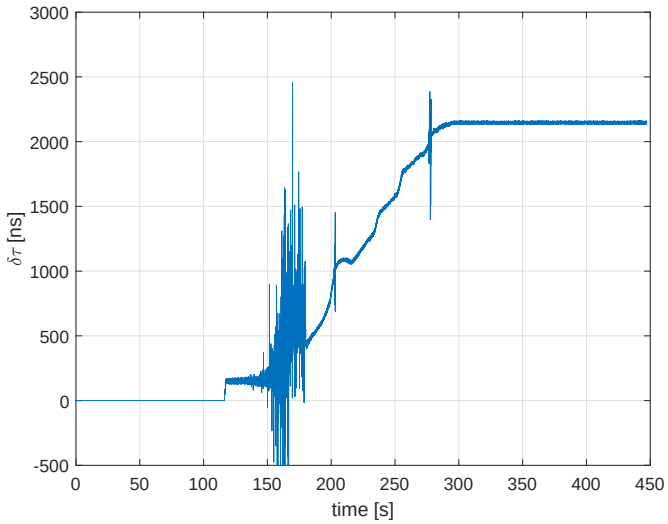

Fig. 7.10 Delay difference $\delta\tau$, between Unit1 and Unit0 for satellite 6, TEXBAT ds3, where the attack starts at 110 seconds.

Unit0. After 100 s of pull-off time, Unit1 estimates the correct delay difference of the real signal as ≈2000 ns, which corresponds to the 600 m of error introduced by the spoofing attack, as stated in [38].

In Figs. 7.11 and 7.12, the estimated amplitudes $\hat{a}_0$ and $\hat{a}_1$ and the estimated phase differences $\hat{\theta}_0$ and $\hat{\theta}_1$ are presented for Unit0 on the top panel of each figure and Unit1 on the bottom one. For the amplitude, we observe that the final values correspond to $\hat{a}_0 = 0.35$ and $\hat{a}_1 = 0.28$; the difference between them is close to the 1.3 dB of spoofing advantage set for the matched-power scenario. Observing these values, we know that Unit0 is tracking the spoofing signal, and Unit1 is tracking the satellite signal, because of the higher amplitude level in Unit1.

In this case, given that the detection algorithm, as shown in Fig. 7.5, always assumes that the spoofing signal is present in Unit1, the first stage of the detection would fail, but the second step will detect the spoofing signal because the signal processed by Unit1 will not change its power while the relative delay is changing. No matter in what stage of the algorithm the spoofing decision is made, once $D_i(t) = 2$, we can assume that the spoofing signal is the one with the higher power, because this is needed to take control of the receiver. In such a case, in order to mitigate the effects of the spoofing attack, information from the correct unit needs to

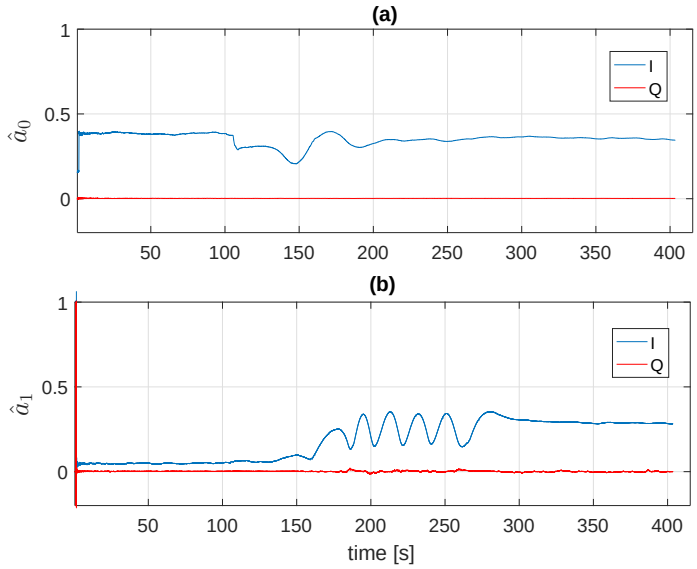

Fig. 7.11 Amplitude estimation for Unit0 (a) and Unit1 (b) for TEXBAT ds3, where the spoofing attack starts at time instant 110 s. In blue, the in-phase amplitude estimation, and in red, the quadrature estimation are shown.

be used for pseudorange computation, as shown in Fig. 7.4, as well as in the PLL of Fig. 7.1, where $\hat{S}_{b,0}$ should be adjusted accordingly.

For the spoofing detection, it can be seen that Unit1 starts following the signal that is being pulled-off after it has a $\delta\tau$ of at least 300 ns, at around 150 s of the dataset. Having a small bandwidth of the front-end will cause the detection to have a delay with respect to the time when the spoofer actually starts modifying the PVT solution, due to the rounding effect in the correlation function. Nevertheless, as demonstrated in [17], this latency can be improved using a better product between front-end bandwidth and the time of chip ($B \cdot T_c$).

These results show the ability of the ECADLL to detect and track the spoofing signal. It is important to notice that given the many possible configurations of spoofing attack scenarios and the many different parameters of the ECADLL architecture, making a statistical analysis in terms of detection probability is a difficult task left for further investigation.

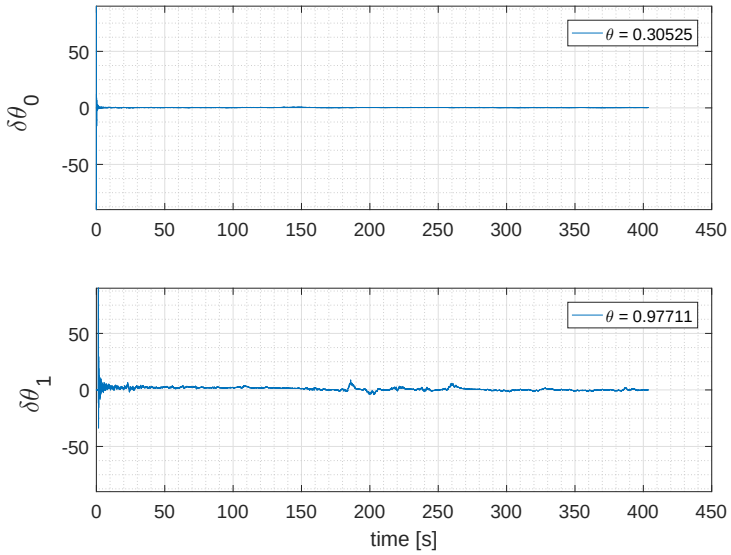

Fig. 7.12 Phase estimation for Unit0 (a) and Unit1 (b) for TEXBAT ds3.

7.4 Results using the spoofing detection algorithm

After observing the correct operation of the ECADLL for the estimation of the signal parameters using the TEXBAT dataset in Section 7.3, in this section, we present the results for the spoofing detection algorithm, along with a comparison versus the SQM technique. Numerical results highlighting the detection capabilities and the improvement brought by the inclusion of RM are also presented.

7.4.1 Detecting an evolved static matched-power time push attack

In this section, we present the results obtained by processing scenario ds7, that is explained in [35] and in Appendix A.1. In Fig. 7.13, we observe how the ECADLL plus RM detector is able to identify the spoofing presence around 120 s. After that, it loses the lock and recovers it once the signals are being pulled apart by the spoofing attack.

In Figs. 7.14 and 7.15, we observe the correct estimation of the delay difference δ_τ and the amplitude estimation that follows the description presented in [35]. We

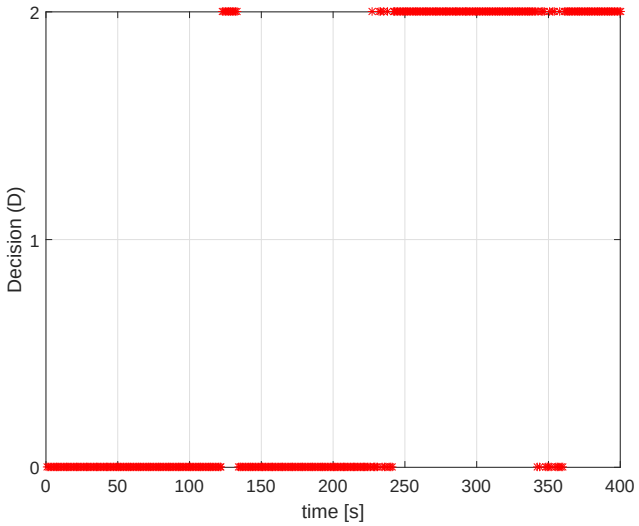

Fig. 7.13 Decision of the spoofing detection algorithm for TEXBAT ds7, where the spoofing attack starts at time instant 110 seconds, and the delay is modifyied starting from time instant 150 seconds.

can observe how the relative difference in amplitude decreases, as the spoofing signal pushes off the real signal and its modular amplitude is reduced.

7.4.2 Comparison between ECADLL and SQM

In this Section, we compare the detection performance of the ECADLL vs. β as presented in Chapter 4 and using the parameters in Table 4.2. Fig. 7.16 shows the decision made by SQM and ECADLL for TEXBAT scenario ds3. In blue, we observe the ECADLL decision. In yellow we observe the decision taken by the β metric, when using the data re-transmitted to the SIGE front-end and in red we see the decision shown in Fig. 4.14. First of all, we notice the big impact that the front-end filter has when using the SQM. We can see that in both scenarios, β is able to detect the distortions at the same time as the ECADLL, but the spoofing classification is delayed when using the SIGE front-end. Also we can see how, in this case, the duration of the detection is reduced to a limited window. Observing the ECADLL and β using the original TEXBAT datasets we see that the detection is comparable between them in terms of latency and duration for scenario ds3. This is an important result because it gives the proposed algorithm a measure of reliability, in terms of detection and classification, and can be then used as a mitigation technique.

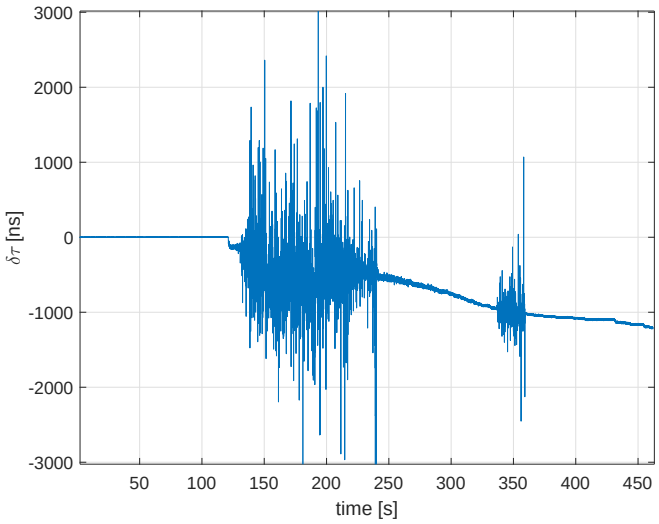

Fig. 7.14 Delay difference estimation $\delta\tau$, for satellite 13, processing TEXBAT ds7.

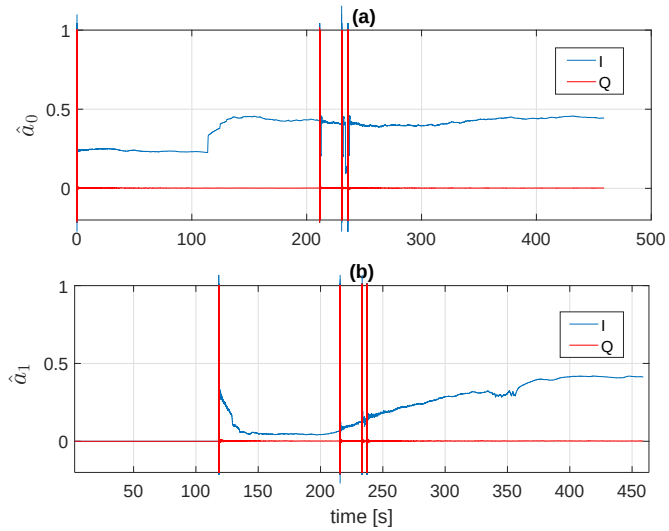

Fig. 7.15 Amplitude estimation for Unit0 (a) and Unit1 (b), for satellite 13, processing TEXBAT ds7. In blue, the in-phase amplitude estimation, and in red, the quadrature estimation are shown.

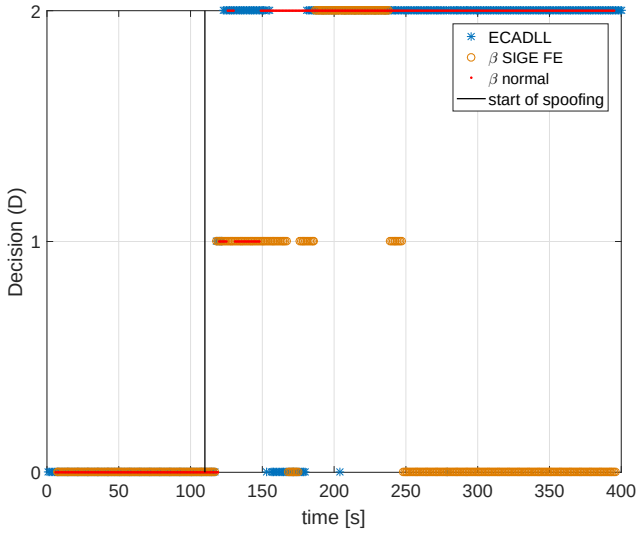

Fig. 7.16 A comparison of the impairment detection using PRN number 6 for the static dataset, TEXBAT ds3, where the spoofing attack starts at 110 s. In blue, the decision made by ECADLL is shown, in yellow, the one using β with the retransmitted data and, in red, the decision taken by β by means of the original dataset.

In the case of a dynamic scenario, we used TEXBAT ds6. The detection results, shown in Fig. 7.17, demonstrate that similar behavior in both static and dynamic scenarios are obtained for the techniques. In this case, the ECADLL presents a delay of around 30 seconds, with respect to the normal β detection, due to the smaller front end bandwidth and the same delay in interference detection is observed for β, in the yellow trend. After 150 s into the test, the ECADLL architecture reaches the steady state of the feedback and starts tracking the satellite signal on Unit1. When using the SIGE front-end, β is also able to detect the spoofing attack, but only for a limited duration of the attack, between 170 s and 290 s, similar to what was observed in Fig. 7.16. At about 50 s in Fig. 7.17, we can observe that a small impairment was detected, and we believe this to be a multipath signal present in the dynamic scenarios of TEXBAT as for all three cases the event is detected.

Comparing the behavior of SQM and ECADLL in Figs. 7.16 and 7.17, we observe how ECADLL has similar detection capabilities to the SQM when the LOS and spoofing signal are close to each other. The ECADLL is able to maintain the detection throughout the test, even when the two signals are well separated. This is not necessarily the case for all SQM configuration as was observed in Figs. 7.16 and 7.17, and was discussed in Section 4.5.

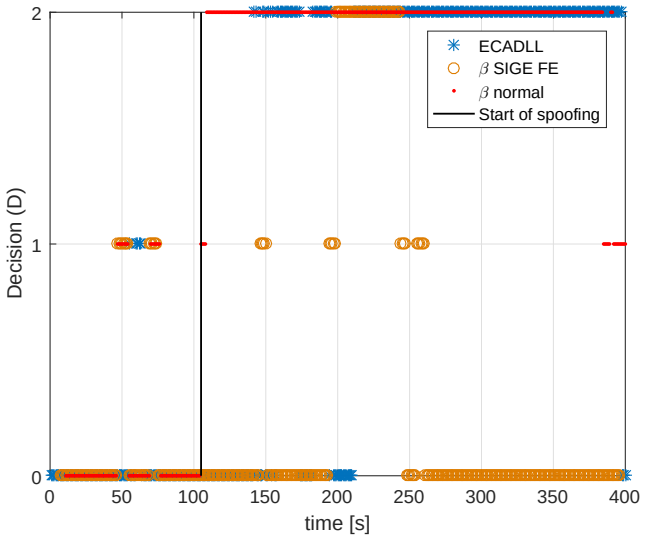

Fig. 7.17 Comparison of the impairment detection using PRN number 15 for the dynamic dataset, TEXBAT ds6, where the spoofing attack starts at around 110 s. In blue, the decision made by ECADLL is shown, in yellow, the one using β with the retransmitted data and, in red, the decision taken by β by means of the original dataset

With the results presented in this Section, we feel confident that the classification algorithm shown in Fig 7.5 can be effectively used for spoofing detection, and will be the base used for mitigation of the spoofing events in Section 7.5.

7.4.3 Numerical results for detection and latency

In this Section, we provide numerical results for the ECADLL algorithm. These results were obtained by processing four datasets with spoofing signals from the TEXBAT and three different clean datasets, one from the clean scenario in the TEXBAT and the other two obtained by means of a moving vehicle in open sky environment in Turin, Italy, described in Appendix A.3. This is not a comprehensive set of datasets, containing all possible configurations of spoofing signals, but we believe that they provide very insightful initial results for demonstrating the capabilities of ECADLL architecture, the detection algorithm and its future potential.

First, we present a confusion matrix including all of the satellites for the different scenarios that were tested and how they were classified by the algorithm. A confusion matrix can be a useful tool to describe the performance of a classification algorithm, such as the one proposed for the ECADLL.

Table 7.1 Confusion matrix for the two types of datasets processed.

		Predicted Outcome		
		Spoofed	**Impairment**	**Clean**
True	Spoofed satellite	22	0	2
	Clean Scenario	0	0	18

In Table 7.1, we can observe the confusion matrix and how the different satellites were classified. For the spoofed scenarios, we observed six satellites for each of the four datasets, obtaining a possible of 24 spoofed satellites. The ECADLL algorithm classified 22 of them correctly and missed the detection of two, not detecting correctly one satellite in each of scenarios ds6 and ds7. We believe that these misdetections were caused by the effects of low C/N_0 that these satellites had for this configuration of the re-transmission. The ECADLL architecture was not able to distinguish clearly the external signal that was present in the correlation domain. On the other hand, no false alarms were present during the clean scenarios, and none of the satellites were wrongly classified.

Following the confusion matrix, which summarizes the detection capabilities of the technique, we will now focus on the improvements that the introduction of RM brought to ECADLL. Two main features were affected by RM, and they will be defined as Computational Time (CT) and Detection Latency (DL). CT will be defined as the time that the algorithm takes to process a single satellite of one scenario of the TEXBAT datasets, and DL is the time difference between the beginning of the push-off phase of the spoofing signal and the first detection of the algorithm, either as an impairment or spoofer. Of course, these values will change between configurations of the receiver and of the spoofing attack, computer implementation and characteristics of the datasets. Therefore, they should not be considered as standalone values, but we can obtain useful information if we compare the same dataset while ECADLL is using RM or not. In Table 7.2, we can observe the numerical improvement for each of the different datasets.

If we look at the Computational Time Improvement (CTI), defined as:

$$CTI = (CT_{base} - CT_{RM})/CT_{base} \tag{7.15}$$

Table 7.2 Numerical results for improvement of detection delay and computational load. Legend: DL = Detection Latency. CT = Computational Time. DLI = Detection Delay Improvement. CTI = Computational Time Improvement.

	Base ECADLL		**ECADLL w\RM**		**Improvement**	
	DL_{base} **(s)**	CT_{base} **(min)**	$DL_{\mathbf{RM}}$ **(s)**	CT_{RM} **(min)**	**DLI** (%)	**CTI** (%)
clean	-	35	-	21	-	40
ds3	27	38	17	35	37	7.9
ds6	33	36	17	33	48.5	8.3
ds7	80	41	26	33	57.5	19.5
ds4	98	38	93	30	5.1	21

we can see how the biggest improvement was obtained when the clean scenario was processed, this means when no spoofer was present. This result is intuitive because when no spoofer is present, the use of RM for the activation of the secondary units will prevent ECADLL from having the two units active at all times. The presence of the Unit1 by itself represents a high burden on the computational load of the receiver, given that it can be comparable to having twice as many channels in tracking and, thus, twice as many correlators.

The CTI, of scenarios with the presence of spoofing signals, is not as large because while the spoofer signal is present, both configurations are running two units for each channel. This means that the improvement will be weighted by the amount of time the spoofing signals are not present in the dataset, which for ds3 and ds6 is about 1/4 and for ds7 and ds4 is around 1/3 of the total time.

Eventually, we focus our attention on the Detection Latency Improvement (DLI), defined as:

$$DLI = (DL_{RM} - DL_{base})/DL_{base} \tag{7.16}$$

where the values of DL_{RM} and DL_{base} are obtained by assuming that the spoofer starts modifying its delay at time 110 s of each dataset as reported in [38]. For scenarios ds4 and ds6, which are modifying the position according to their selected pattern, each satellite will be modified with a different delay, unknown to us. According to our analysis, the modification of the delay for ds6 occurs closer to the 110 s mark, than the modification for ds4.

It is interesting to notice the significant improvement in latency obtained when considering ds7. Given the slower rate at which the spoofer modifies the delay with

respect to the other scenarios, we get a longer detection time when only relying on the tracking lock of Unit1. When using RMs, we see that we have a much lower latency, and we are able to flag the presence of impairments in a quicker way.

Values in Table 7.2 give us the numerical results on how the architecture performance was improved by the introduction of the RM as a simple additional tool for the monitoring block. It is also important to notice that the detection latency is not necessarily referring to a spoofer classification by the algorithm, but it represents the initial warning that the RM gives to the receiver for it to not trust the signal coming from that satellite, until it can be correctly classified or corrected by means of ECADLL.

7.5 Mitigation of the spoofer effects

Finally, we demonstrate the capabilities for spoofing mitigation of the ECADLL. In order to do this, we will use the ds4 scenario as an example of the results that can be obtained by mitigating the spoofing effects. The mitigation process used for these results relies on the classification algorithm presented in 7.2. The mitigation will be performed by switching the information between Unit0 and Unit1, if Unit1 is declared as being the satellite signal.

The first step of the mitigation process comes from the decision $D_i(t_k)$, taken for each satellite. After a spoofing presence has been declared we need to decide whether the spoofing signal is being tracked by Unit0 or Unit1. As stated before, the spoofing signal needs to have a larger amplitude than the satellite signal. Thus by observing the amplitude value $\hat{a}$ for each unit, we flag the unit with the larger amplitude as containing the spoofing signal.

The receiver computes the navigation solution using information in Unit0, so we need to make sure that the satellite signal is been tracked by Unit0. That means that if the Unit0 of a channel i is declared by D_i as containing spoofing signals, the information of Unit1, that will contain the satellite signal, needs to be switched with that of Unit0. In order to proceed with the switch, we first need to make sure that the signals tracked by Unit0 and Unit1 are separated enough so the switch can be done without losing track of the signals. To assure this, similar to what was proposed in Section 6.4, we perform a check on the estimated delay difference $\delta\tau$. If $\delta\tau > 0.5\mu$s

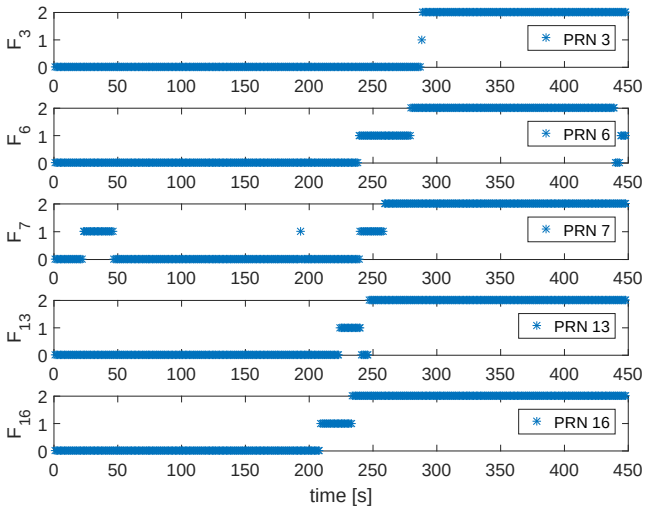

Fig. 7.18 Flag for each satellite for scenario ds4, where the spoofing attack starts at around 110 seconds, but each PRN is affected independently. Mitigation starts at around 230 seconds, and after 280 seconds the solution is fully mitigated

we perform the switch of the units' information. In this way, the mitigation will be done as soon as a channel meets the required conditions and the process will eventually remove all spoofed signals from the navigation solution.

Summarizing, every satellite in the PVT solution will be flagged as:

$$F_i(t_k) = \begin{cases} 2 & \text{Mitigated} \\ 1 & \text{Spoofed} \\ 0 & \text{Normal} \end{cases} \tag{7.17}$$

In Fig. 7.18 we observe the flag given to each satellite tracked in the example ds4. We observe how all of the satellites are gradually being mitigated and after 280 s, the full PVT solution will be free of spoofing presence.

In Fig. 7.19 we can observe the effect of the mitigation in the navigation domain, namely in the x, y and z axes. In red, the position changes of the clean dataset are shown as reference, and we observe minimum variations on it. In blue, the variations presented under spoofing attack are shown and we can observe high variations, especially on y and z. In green we present the results of the mitigated solution. For

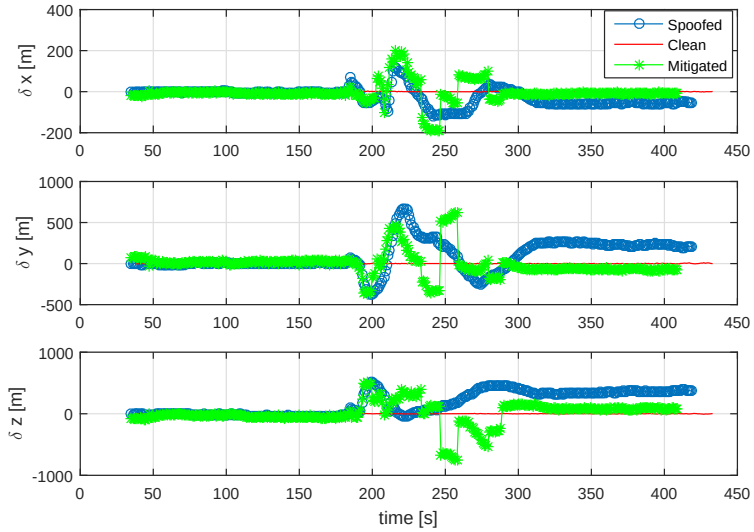

Fig. 7.19 x, y and z mitigation in term of the coordinate errors results from the PVT using TEXBAT ds4

the mitigated plot we observe that after 280 seconds, the mitigated solution is closer to its original values, once the mitigation is completed on all satellites.

The mitigationeffects can be observed from the results depicted in Figs. 7.19 and 7.18. At the beginning of the spoofing attack, the mitigated solutions follows closely the regular solution. After some time, in this example ≈ 50 seconds after the beginning of the mitigation, the mitigation is completed in all the satellites and the solution errors are reduced. This feature can be observed in Fig. 7.20, where the East-North-Up plot is shown with different colors for the different stages of the mitigation.

We can observe how the red dots, which in this case represent the full mitigation solution, are bounded to errors lower than 10 meters East and 50 meters North, while for the spoofed solution, the errors go up to 200 meters East and 500 meters North. It is normal to expect that the solution will not be as precise as the one obtained with the clean scenario, given that the signal will be disrupted by the presence of the spoofer and the C/N_0 is lower for all the satellites due to the extra noise added by the spoofing attack. Nevertheless, the solution is not controlled any more by the spoofer and the errors are lower.

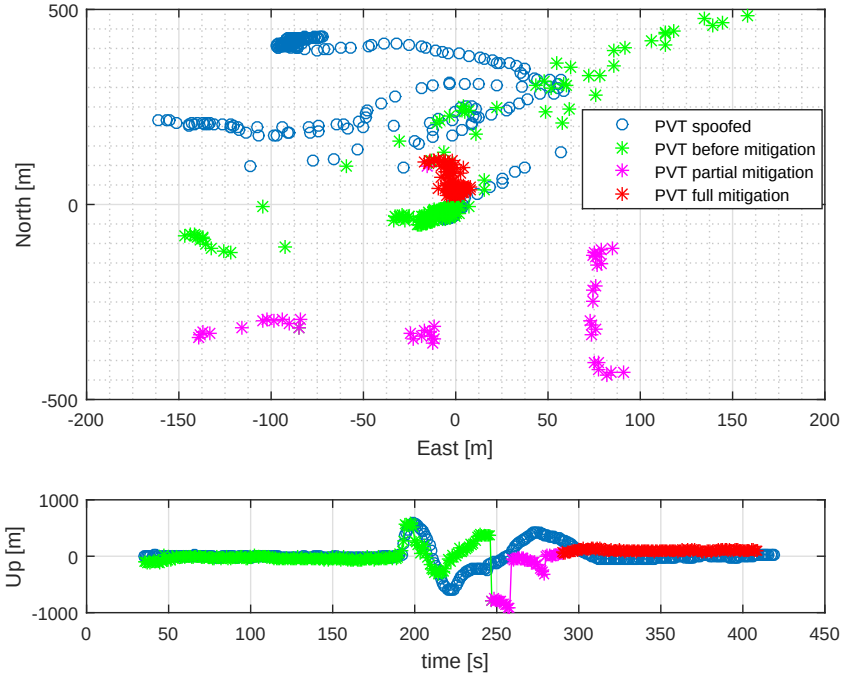

Fig. 7.20 East - North - Up mitigation results in the navigation domain for TEXBAT ds4 scenario

Table 7.3 Mean, standard deviation and maximum 3D rms error in meters, generated from the spoofed scenario and from the mitigated case

3D rms error	mean	std	max
Spoofed (from 110 s)	405	108	678
Mitigated (from 110 s)	246	208	975
After Full Mitigation (from 288 s)	116	20	173

Finally in Table 7.3 we can observe the statistics on the 3D rms error obtained from each case after the spoofing attack started after time instant 110 s. In the table, the first row represents the statistics on the errors for the spoofed scenario, as shown in blue dots in Fig. 7.20, the second row presents the overall mitigated solution after the spoofing attack starts, thus combining green, purple and red dots from Fig. 7.20; and the last row presents only the statistics when the mitigation on all satellites is performed, thus taking in consideration only the red dots in Fig. 7.20. We can observe how the mean error is considerably reduced from the spoofer to the mitigated case, although it still seem to be a bias in the solution after the mitigation has being performed completely.

It is also important to remark that the used PVT algorithm is a basic least square solution and using a more evolved version, e.g. a Kalman filter solution, could help obtaining a better accuracy of the estimated position as it was observed for the TJ algorithm in Chapter 6.

Summary and Conclusions

Throughout this thesis, we have discussed several anti-spoofing techniques, validating their performances against different datasets and introducing improvements and additional checks that can be integrated to the procedures in order to improve spoofing detection capabilities.

As well known, the GNSS community is paying great attention to all aspects related to the security of GNSS-based applications. At the signal level, apart from the degradation due to the environment, such as multipath reflections, foliage, ionospheric effects, etc., intentional sources of disturbances might be present, such as jamming or spoofing signals, which deliberately attempt to disrupt the correct signal reception.

In this thesis, the author concentrated on the spoofing attacks for GNSS receiver, with particular attention to signal processing techniques used for spoofing detection purposes. Within this context, significant contributions on the design and validation of detection and mitigation algorithms, based on the use of four different signal processing techniques, are introduced in this thesis. Three of the four techniques presented in this thesis are based on previous work, and the contribution of the author was to update the detection algorithm as well as to introduced new checks to reduce false alarms due to external elements, such as multipath and RFI. The last technique presented, is a novel technique, designed within this doctoral program in order to integrate the research areas of the involved students and perform mitigation of the received signals using a signal decomposition approach.

All the techniques were validated against the same set of data which allows for a more systematic comparison of performance and a reinforcement of the results obtained. A thorough analysis, validation and testing of the techniques against this anti-spoofing battery have not been previously done.

The literature proposes several anti-spoofing techniques, based on a wide variety of different approaches. Among them, SQM techniques are methods based on the monitoring of the GNSS correlation function. They employ proper tests, i.e. the ratio test, to detect asymmetries in the correlation function. These techniques are very powerful and able to detect very small distortions. At the same time, since the presence of spoofing signals can affect the correlator output in a way similar to that of multipath components, SQM metrics might not be able to discriminate between these impairments.

Chapter 3 focused on the general description of the SQM techniques and presented a detection algorithm to be implemented in a receiver which is a novel contribution. With this detection algorithm, the receiver is able to compute a threshold by means of the false alarm probability, and it then compares all the values of the metric M inside a DW. The algorithm finally aims to make a decision based on the percentage of values above said threshold. Results are presented by means of the TEXBAT datasets and we observe how the technique is able to detect distortions of the correlation function in case of spoofing attacks. All satellites were detected in the static scenario, while in the dynamic case, one of the satellites was not detected. Using the SQM provides a transient indicator of the spoofer presence, only detecting a subset of the possible delay and amplitude, and not categorically all. This could be solved by using a scanning correlator that observes the different delays and gives a warning once a signal is detected outside the correlation peak' as an example of future work. The detection capabilities of the SQM metric M are improved by the introduction of the β metric, as shown in the results of Chapter 4.

Chapter 4 presented a new multidimensional metric, also based on the ratio test metric M. It is able to take the decision on the presence or absence of distortions in the received signal and contemporary discriminate between spoofing and environmental effects. It exploits the fact that, typically, a spoofing attack tends to be continuous over time, while a multipath event might be characterized by faster dynamics, especially in dynamic urban scenarios. Moreover, the metric considers that, typically, in real scenarios, multipath signals do not affect contemporary all the satellites in view, while, under a spoofing attack, it is likely to detect anomalies in the same time intervals, for a big subset of the tracked signals. The feasibility of thoroughly combining very different parameters into a decision making metric has been proved and validated, and it is able to detect spoofing attacks and discriminate them from the distortions generated by environmental effects.

When considering the SQM, all the metrics were based on the in-phase branch of the receiver. Nevertheless, for improving the detection of more advanced spoofers able to freely rotate the phase of the signal, M and, consequently, β could be easily duplicated on the quadrature branch of the tracking loop. By means of the information available in the quadrature branch of the DLL, the spoofing detection using the SQM metric becomes independent of the phase of the spoofer signal.

In Chapter 5 we have presented a technique able to detect overpowered and matched powered spoofing attacks, by means of the joint observation of the power measurements, via the control of the AGC and the observation of asymmetries in the correlation function, via an SQM technique. Two metrics were defined, and their nominal statistics were computed using data from six different WAAS stations spread across the United States. From these stations, over 24 hours of correlator data were used for assessment of metric M, and over 120 hours of AGC gain information were used for the assessment of the G_{AGC} metric. All this data was obtained by means of a Novatel G-II receiver. By processing the TEXBAT datasets with the receiver, we demonstrated the detection capabilities of both metrics for overpowered and matched power scenarios using a COTS receiver.

Additionally in Chapter 5, we introduce a new check to reduce false alarms likely due to interference presence. These effects were observed when we analyzed the data from the WAAS stations. These false alarms can be discriminated by means of the cross-observation of the curve of G_{AGC} vs C/N_0. Controlled simulations were performed in order to identify correctly the behavior of the AGC with respect to the C/N_0, and the obtained thresholds were applied to the spoofing detection algorithm. With the observation of the G_{AGC} vs C/N_0 curve we were able to accurately eliminate the false alarms from the WAAS stations, caused by RFI interference. Using the different aspects of the receiver, the technique presented in Chapter 5 is able to correctly detect and identify spoofing attacks, while maintaining a low probability of false alarm due to the presence of interference. At the end of the Chapter, the complete algorithm that would need to be implemented in order to obtain protection from matched-power and overpowered spoofing attack, and to lower false alarms due to RFI signals and multipath events is presented. With this we concluded the detection part of the thesis where we focused on techniques that are easily implemented in the receiver, but at the same time provide powerful detection capabilities.

In Chapter 6 we presented the TJ algorithm, which is able to detect spoofing attacks and mitigate its effects in a GNSS receiver. Moreover, in parallel to these operations, the algorithm tries to guarantee the continuity and reliability of the system by means of a Kalman filter using only Doppler measurements. These Doppler measurements are more reliable during pseudorange-based attacks. The combination of several techniques like LASSO, Kalman filter and time jumping procedure at different levels, increases the complexity of the system, but allows to have a more robust receiver that is able to mitigate the negative effects of a spoofing attack. The operation of the TJ algorithm was demonstrated by means of real test scenarios, both in static and dynamic modes, observing how the position error is considerably reduced. As for the SQM techniques, in the TJ algorithm, the signal processing part, LASSO, SQI and BDE should be applied to both branches (I and Q) of the receiver, in order to take into account possible attacks that use relative phase between the LOS and the spoofer signal. On the list of future work, remains the testing and improvement of this algorithm by means of new datasets containing additional configurations of spoofing attacks, with the aim of optimizing the performance in terms of computational effort.

Finally in Chapter 7, a spoofing detection algorithm based on the use of ECADLL was introduced. The algorithm is able to identify spoofing attacks when different conditions are met, based on the expected behavior of multipath reflections and on the dynamics of the receiver. The ECADLL architecture along with the spoofing detection algorithm were tested against the TEXBAT datasets, demonstrating that the technique is able to detect the spoofing attack scenario and that it is able to provide accurate estimation of the parameters of the spoofing signal. The detection algorithm presented in this work is based on a single satellite observation. Looking at more satellites simultaneously, using a parallel architecture of the algorithm, could improve the decisions and help identify more easily the spoofing attack.

The ECADLL version presented in the thesis, including the spoofing detection algorithm and the ratio metrics in the monitoring block, has overall good detection capabilities and low false alarms, while improving over the original ECADLL in the computational burden and on the detection latency. On the other hand, a careful definition of the thresholds for the RM is important, in order to turn on the units when it is needed and obtain optimal performances. Additionally the capabilities of mitigation of the spoofing effects were presented with promising results. Tuning of the mitigation algorithm as well as the PVT computation techniques could improve

the performance for mitigation. By using a front-end with a higher filter bandwidth and more bits of resolution, the ECADLL receiver would improve its detection latency and the delay estimation would be more accurate due to the sharper peak.

Similar to what was discussed for the SQM and TJ algorithm, the ECADLL is not able to defend the receiver in the cases of overpowered attacks, where the vestigial receiver signal is buried under the noise levels introduced by the spoofer. Fortunately, these types of attacks are easily detectable by using in-band power measurements as observed in Chapter 5 even if no mitigation would be possible in these cases.

Overall the introduced techniques present good detection capabilities and each one of them is designed with the practical receiver implementation as a goal. All the techniques are feasible as stand alone signal processing techniques, with the abilities of detection and mitigation of the spoofing attack.

References

[1] Sige front end hardware module, 2008. Available on line: https://ccar.colorado.edu/gnss/.

[2] UT Austin Researchers Successfully Spoof an $80 million Yacht at Sea, 2013. Available on line: http://www.utexas.edu/news/2013/07/29/ut-austin-researchers-successfully-spoof-an-80- million-yacht-at-sea/.

[3] D. M. Akos. Who's Afraid of the Spoofer? GPS/GNSS Spoofing Detection via Automatic Gain Control (AGC). *Journal of the Institute of Navigation*, 59(4), Winter 2012.

[4] K. Ali, X. Chen, and F. Dovis. On the use of multipath estimating architecture for spoofer detection. In *2012 Int. Conference on Localization and GNSS (ICL-GNSS)*, pages 1–6, June 2012.

[5] K. Ali, E. Garbin Manfredini, and F. Dovis. Vestigial signal defense through signal quality monitoring techniques based on joint use of two metrics. In *2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014*, Monterey, CA, May 2014.

[6] J. A. Avila-Rodriguez, G. W. Hein, S. Wallner, J. Issler, L. Ries, L. Lestarquit, A. Latour, J. Godet, F. Bastide, T. Pratt, et al. The mboc modulation: the final touch to the galileo frequency and signal plan. *Navigation*, 55(1):15–28, 2008.

[7] F. Bastide, D. M. Akos, C. Macabiauand, and B. Roturier. Automatic gain control (agc) as an interference assessment tool. In *Proceedings of the 16th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS/GNSS 2003)*, Portland, OR, September 2003. Institute of Navigation.

[8] M. Berardo and L. Lo Presti. GNSS multipath detector based on linear adaptive filter. In *Proc. of the 28th Int. Tech. Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2015)*, pages 3077–3083, Tampa, FL, 2015.

[9] M. Berardo and L. Lo Presti. On the use of a signal quality index applying at tracking stage level to assist the raim system of a gnss receiver. *Sensors*, 16(7): 1029, 2016. ISSN 1424-8220. URL http://www.mdpi.com/1424-8220/16/7/1029.

[10] M. Berardo, E. Garbin Manfredini, L. Lo Presti, and F. Dovis. A spoofing mitigation technique for dynamic applications. In *2016 8th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*, Noordwijk, The Netherlands, December 2016.

[11] T. Blumensath and M. E. Davies. Iterative thresholding for sparse approximations. *J. of Fourier Analysis and Applications*, 14(5-6):629–654, 2008.

[12] D. Borio, F. Dovis, H. Kuusniemi, and L. Lo Presti. Impact and detection of gnss jammers on consumer grade satellite navigation receivers. *Proceedings of the IEEE*, 104(6):1233–1245, June 2016. ISSN 0018-9219. doi: 10.1109/JPROC.2016.2543266.

[13] K. Borre, D. M. Akos, N. Bertelsen, P. Rinder, and S. H. Jensen. *A software-defined GPS and Galileo receiver: a single-frequency approach*. Springer Science & Business Media, 2007.

[14] S. Boyd and L. Vandenberghe. *Convex optimization*, pages 291–302. Cambridge University Press, Cambridge, UK, 2004.

[15] R. G. Brown and P. Y. C. Hwang. *Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions*, volume 1. John Wiley and Sons, New York, NY, third edition, 1997.

[16] A. Cameron. New beidou tmboc signal tracked; similar to future gps l1c structure. *GPS World*, October 215. URL http://gpsworld.com/new-beidou-tmboc-signal-tracked-similar-to-future-gps-l1c-structure/.

[17] X. Chen and F. Dovis. Enhanced cadll structure for multipath mitigation in urban scenarios. In *Proceedings of the 2011 International Technical Meeting of The Institute of Navigation*, pages 678–686, San Diego, CA (USA), January 2011. URL http://porto.polito.it/2496696/.

[18] X. Chen, F. Dovis, M. Pini, and P. Mulassano. Turbo architecture for multipath mitigation in global navigation satellite system receivers. *Radar, Sonar Navigation, IET*, 5(5):517–527, June 2011.

[19] S. Daneshmand, A. Jafarnia-Jahromi, A. Broumandan, and G. Lachapelle. A low complexity gnss spoofing mitigation technique using a double antenna array. *GPS World*, 22(12):44–46, 2011.

[20] H. V. de Castro, G. van der Maarel, and E. Safipour. The Possibility and Added-value of Authentication in future Galileo Open Signal. In *Proceedings of the 23th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2010)*, Portland, OR, September 2010.

[21] H. V. de Castro, G. van der Maarel, and E. Safipour. GNSS interference detector based on Chi-square Goodness-of-fit test. In *2012 6th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing, (NAVITEC)*, pages 1–6, 2012.

[22] A. J. Van Dierendonck, P. Fenton, and T. Ford. Theory and Performance of Narrow Correlator Spacing in a GPS Receiver. In *Proceedings of the 1992 National Technical Meeting (NTM) of The Institute of Navigation*, pages 115–124, San Diego, CA, January 1992.

[23] F. Dovis. *GNSS Interference Threats and Countermeasures*. Artech House, Norwood, MA, 2015.

[24] F. Dovis, X. Chen, A. Cavaleri, and K. Ali. Detection of spoofing threats by means of signal parameters estimation. In *Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011)*, Portland, OR, September 2011.

[25] Global Positioning System Directorade Systems Engineering and Integration. Interface specification is-gps-200, 2014. URL http://www.gps.gov/technical/icwg/IS-GPS-200H.pdf.

[26] E. Falletti, M. Pini, and L. Lo Presti. Are carrier-to-noise algorithms equivalent in all situations? *INSIDE GNSS*, 5(1):20–27, 2010. URL http://www.insidegnss.com/node/1826.

[27] G. W. Hein and F. Kneissl and J. Avila Rodriguez and S. Wallner. Authenticating GNSS: Proofs Against Spoofs. Part II. *Inside GNSS*, 2(6):71–78, Sept./Oct 2007.

[28] GPS World staff. US Coast Guard issues GPS jamming alert, 2016. URL http://gpsworld.com/us-coast-guard-issues-gps-jamming-alert/. GPS World.

[29] GPS World staff. Firmware update for u-blox M8 GNSS receiver adds Galileo, 2016. URL http://gpsworld.com/firmware-update-for-u-blox-m8-gnss-receiver-adds-galileo. GPS World.

[30] A. Grant and P. Williams. GNSS Solutions: What is the effect of GPS jamming on maritime safety? *Inside GNSS*, 4(1), Jan/Feb 2009.

[31] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer-Verlag, New York, NY, 2 edition, 2009. URL http://www-stat.stanford.edu/~tibs/ElemStatLearn/.

[32] G. W. Hein, J. A. Avila-Rodriguez, S. Wallner, A. R. Pratt, J. Owen, J. Issler, J. W. Betz, C. J. Hegarty, S. Lenahan, J. J. Rushanan, A. L. Kraay, and T. A. Stansell. Mboc: The new optimized spreading modulation recommended for galileo l1 os and gps l1c. In *2006 IEEE/ION Position, Location, And Navigation Symposium*, pages 883–892, April 2006. doi: 10.1109/PLANS.2006.1650688.

[33] L. Heng, D. B. Work, and G.X. Gao. Cooperative GNSS Authentication. Reliability from Unreliable Peers. *Inside GNSS*, 8(5):70–75, September/October 2013.

[34] L. Huang and Q. Yang. Gps spoofing: Low-cost GPS simulator, 2015. URL https://media.defcon.org/DEF%20CON%2023/DEF%20CON%2023%20presentations/Lin%20Huang%20&%20Qing%20Yang/DEFCON-23-Lin-Huang-Qing-Yang-GPS-Spoofing.pdf. Presented during DEFCON 23rd conference by the Unicorn Team.

[35] T. E. Humphreys. Texbat data sets 7 and 8, 2015. Available on line: http://radionavlab.ae.utexas.edu/research/50-projects/radionavigation-datasets/289-texas-spoofing -test-battery-texbat.

[36] T. E. Humphreys, B. M. Ledvina, L. M. Psiaki, B. W. O Hanlon, and P. M. Kintner. Assesing the spoofing threat: development of a portable gps civilian spoofer. In *Proceedings of the 21th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2008)*, Savannah, GA, September 2008.

[37] T. E. Humphreys, B. A. Ledvina, M. L. Psiaki, B. W. O Hanlon, and Jr. P. M. Kintner. Assessing the spoofing threat. *GPS World*, 20(1):28–38, January 2009.

[38] T. E. Humphreys, J. A. Bhatti, D. P. Shepard, and K. D. Wesson. The Texas Spoofing Test Battery: Toward a Standard for Evaluating GPS Signal Authentication Techniques. In *Proceedings of the 25th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2012)*, Nashville, TN, September 2012.

[39] M. Irsigler. *Multipath propagation, mitigation and monitoring in the light of Galileo and the modernized GPS*. PhD thesis, Universität der Bundeswehr München, 2008.

[40] J. Huang and L. Lo Presti and B. Motella and M. Pini. GNSS spoofing detection: Theoretical analysis and performance of the Ratio Test metric in open sky. *ICT Express*, 2(1):37–40, March 2016.

[41] A. Jafarnia-Jahromi, A. Broumandan, J. Nielsen, and G. Lachapelle. GPS spoofer countermeasure effectiveness based on signal strength, noise power and C/N0 observables. *International Journal of Satellite Communications and Networking*, 30(4):181–191, 2012.

[42] A. Jafarnia-Jahromi, A. Broumandan, J. Nielsen, and G. Lachapelle. GPS vulnerability to spoofing threats and a review of antispoofing techniques. *International Journal of Navigation and Observation*, 2012, June 2012.

[43] J. C. Juang. GNSS spoofing analysis by VIAS. *Coordinates*, VI(1):11–14, January 2011.

[44] R. E. Kalman. A new approach to linear filtering and prediction problems. *J. of Fluids Engineering*, 82(1):35–45, 1960.

[45] E. D. Kaplan and C. J. Hegarty. *Understanding GPS: Principles And Applications*. Artech House, Norwood, MA, 2005.

[46] S. M. Kay. *Fundamentals of Statistical Signal Processing: Detection Theory*, volume II of *Signal Processing*. Prentice Hall, New Jersey, 1998.

[47] J. Klobuchar. Ionospheric effects on gps. *Global Positioning System: Theory and applications.*, 1:485–515, 1996.

[48] H. Kuusniemi, A. Wieser, G. Lachapelle, and J. Takala. User-level reliability monitoring in urban personal satellite navigation. *IEEE Transactions on Aerospace and Electronic Systems*, 43(4):1305–1318, 2007.

[49] B. M. Ledvina, W. J. Bencze, B. Galusha, and I. Miller. An in-line anti-spoofing device for legacy civil gps receivers. In *Proceedings of the 2010 International Technical Meeting of The Institute of Navigation*, San Diego,CA, January 2010.

[50] A. Lemmenes, P. Corbell, and S. Gunawardena. Detailed Analysis of the TEXBAT Datasets Using a High Fidelity Software GPS Receiver. In *Proc. of the 29th Int. Tech. Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2016)*, Portland, OR, September 2016.

[51] H. W. Lilliefors. On the kolmogorov-smirnov test for normality with mean and variance unknown. *Journal of the American statistical Association*, 62(318): 399–402, 1967.

[52] S. Lo, D. De Lorenzo, P. Enge, D. Akos, and P. Bradley. Signal authentication: A secure civil gnss for today. *inside GNSS*, 4(5):30–39, 2009.

[53] E. Garbin Manfredini and F. Dovis. On the use of a feedback tracking architecture for satellite navigation spoofing detection. *Sensors*, 16(12):2051, 2016. ISSN 1424-8220. doi: 10.3390/s16122051. URL http://www.mdpi.com/1424-8220/16/12/2051.

[54] E. Garbin Manfredini, F. Dovis, and B. Motella. Validation of a signal quality monitoring technique over a set of spoofed scenarios. In *2014 7th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*, Noordwijk, The Netherlands, December 2014.

[55] E. Garbin Manfredini, F. Dovis, and B. Motella. Signal quality monitoring for discrimination between spoofing and environmental effects, based on multidimensional ratio metric tests. In *Proceedings of the 28th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2015)*, Tampa, FL (USA), September 2015. Institute of Navigation.

[56] C. E. McDowell. Gps spoofer and repeater mitigation system using digital spatial nulling - us, 2007.

[57] E. McMilin, D. S. De Lorenzo, T. Walter, T. H. Lee, and P. Enge. Single antenna gps spoof detection that is simple, static, instantaneous and backwards compatible for aerial applications. In *Proc. of the 27th Int. Tech. Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2014)*, pages 2233–2242, Tampa, FL, 2014.

[58] E. McMilin, Y.H. Chen, D. S. De Lorenzo, S. Lo, D. Akos, and P. Enge. Field test validation of single-element antenna with anti-jam and spoof detection. In *Proceedings of the 28th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2015)*, Tampa, FL, September 2015.

[59] P. Misra and P. Enge. *Global Positioning System: Signals, Measurements and Performance Second Edition*. Lincoln, MA: Ganga-Jamuna Press, 2006.

[60] P. Y. Montgomery, T. E. Humphreys, and B. M. Ledvina. A multi-antenna defense: Receiver-autonomous gps spoofing detection. *Inside GNSS*, 4(2): 40–46, April 2009.

[61] B. Motella, M. Pini, and F. Dovis. Investigation on the effect of strong out-of-band signals on global navigation satellite systems receivers. *GPS Solutions*, 12(2):77–86, 2008.

[62] J. Nielsen, A. Broumandan, and G. Lachapelle. Spoofing detection and mitigation with a moving handheld receiver. *GPS World*, 21(9):27–33, September 2010.

[63] China Satellite Navigation Office. Beidou navigation satellite system signal in space interface control document, December 2013. URL http://www.beidou.gov.cn/attach/2013/12/26/20131226b8a6182fa73a4ab3a5f107f762283712.pdf.

[64] P. Y. Montgomery and T. E. Humphreys and B. M. Ledvina. Receiver-autonomous spoofing detection: experimental results of a multi-antenna receiver defense against a portable civil GPS spoofer. In *Proceedings of the 2009 International Technical Meeting of The Institute of Navigation*, Anaheim, CA, January 2009.

[65] R. E. Phelts. *Multicorrelator techniques for robust mitigation of threats to GPS signal quality*. PhD thesis, Standford University, Palo Alto, CA. USA, 2001.

[66] R. E. Phelts, T. Walter, and P. Enge. Toward realtime sqm for waas: Improved detection techniques. In *Proc. of the 16 th Int. Tech. Meeting of the Satellite Division of the Institute of Navigation, (ION GPS-2003)*, pages 2739–2749, Portland, OR, September 2003.

[67] M. Pini, M. Fantino, A. Cavaleri, S. Ugazio, and L. Lo Presti. Signal quality monitoring applyed to spoofing detection. In *Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011)*, Portland, OR, September 2011.

[68] M. Pini, B. Motella, L. Pilos, L. Vesterlund, D. Blanco, F. Lindstrom, and C. Maltoni. Robust on-board ship equipment: the TRITON project. In *Proceedings of the 10th International Symposium Information on Ships*, Hamburg, Germany, September 2014.

[69] O. Pozzobon, L. Canzian, M. Danieletto, and A. D. Chiara. Anti-spoofing and open gnss signal authentication with signal authentication sequences. In *2010 5th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*, pages 1–6, Dec 2010. doi: 10.1109/NAVITEC.2010.5708065.

[70] M. L. Psiaki and T. E. Humphreys. GNSS spoofing and detection. *Proceedings of the IEEE*, 104(6):1258–1270, 2016.

[71] M. L. Psiaki, S. P. Powell, and B. W. O Hanlon. GNSS Spoofing Detection. Correlating Carrier Phase with Rapid Antenna Motion. *GPS World*, 24(6): 53–58, June 2013.

[72] S. Pullen and G. Gao. GNSS Jamming in the Name of Privacy. *Inside GNSS*, March/April 2012.

[73] C. Ramirez, V. Kreinovich, and M. Argaez. Why ℓ_0 is a good approximation to ℓ_0: A geometric explanation. *J. of Uncertain Systems*, 7(3):203–207, 2013.

[74] A. Rugamer, M. Stahl, I. Lukcin, and G. Rohmer. Privacy protected localization and authentication of georeferenced measurements using galileo prs. In *2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014*, pages 478–486, May 2014. doi: 10.1109/PLANS.2014.6851406.

[75] European GNSS Gaileo Open Service. Signal in space. inteface control document, 2015. URL https://www.gsc-europa.eu/system/files/galileo_documents/Galileo_OS_SIS_ICD.pdf.

[76] D. Shepard and T. E. Humphreys. Characterization of receiver response to spoofing attacks. In *Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011)*, "Portland, OR", September 2011.

[77] C. Soussen, J. Idier, Junbo Duan, and D. Brie. Homotopy based algorithms for ℓ_0 - regularized least-squares. *IEEE Trans. Signal Processing*, 63(13): 3301–3316, July 2015. ISSN 1053-587X. doi: 10.1109/TSP.2015.2421476.

[78] X. Tang, G. Falco, E. Falletti, and L. Lo Presti. Complexity reduction of the kalman filter-based tracking loops in gnss receivers. *GPS Solutions*, pages 1–15, 2016.

[79] X. Tang, Y. Yang, X. Chen, G. Falco, and E. Falletti. A newly designed tracking loop for power-saving receivers. In *2016 Fourth International Conference on Ubiquitous Positioning, Indoor Navigation and Location Based Services (UPINLBS)*, pages 102–106, Nov 2016. doi: 10.1109/UPINLBS.2016.7809957.

[80] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. ISSN 00359246. URL http://www.jstor.org/stable/2346178.

[81] K. D. Wesson, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys. An evaluation of the vestigial signal defense for civil gps anti-spoofing. In *Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011)*, Portland, OR, September 2011.

[82] K. D. Wesson, M. Rothlisberger, and T. E. Humphreys. Practical cryptographic civil gps signal authentication. *NAVIGATION, Journal of The Institute of Navigation*, 59(3):177–193, 2012.

[83] K. D. Wesson, D. Shepard, and T. E. Humphreys. Straight talk on anti-spoofing. *GPS World*, 23(1):32–39, 2012.

[84] K. D. Wesson, B. L. Evans, and T. E. Humphreys. A combined symmetric difference and power monitoring gnss anti-spoofing technique. In *Proceeding of the 1st IEEE Global Conference on Signal and Information Processing*, Austin, TX, December 2013.

[85] N. A. White, P. S. Maybeck, , and S. L. DeVilbiss. Detection of interference/jamming and spoofing in a DCPS-aided inertial system. *IEEE Transactions on Aerospace and Electronic Systems*, 34(4):1208–1217, 1998.

[86] M. Wildemeersch, E. Cano Pons, A. Rabbachin, and J. Fortuny Guasch. Impact study of unintentional interference on gnss receivers technical report. Technical report, EC Joint Research Centre, Security Technology Assessment Unit, 2011.

[87] M. B. Wilk and R. Gnanadesikan. Probability plotting methods for the analysis for the analysis of data. *Biometrika*, 55(1):1–17, 1968.

[88] C. Xi-jun, X. Jiang-ning, C. Ke-jin, and W. Jie. An authenticity verification scheme based on hidden messages for current civilian gps signals. In *Fourth International Conference on Computer Sciences and Convergence Information Technology, 2009. ICCIT '09*, pages 345–352, Nov 2009.

Appendix A

Description of dataset used

In this Appendix we describe the different datasets used for the assessment of anti-spoofing techniques in the Chapters of the thesis. These sets include the TEXBAT, a set of collection done with a commercial receiver at different WAAS stations and several urban datasets collected in Torino, Italy.

A.1 The Texas Anti-spoofing test battery

The TEXBAT is a collection of datasets released by the University of Texas at Austin, that includes datasets to test anti-spoofing techniques. The TEXBAT is used throughout this thesis to validate the different techniques presented. The thorough description of the TEXBAT datsets can be found in [38], and its updates in [35].

The battery is based on two clean scenarios, one static and one dynamic, recorded in Austin, TX in 2011. The spoofing attacks were performed on top of the clean scenarios with different configurations for each case. All scenarios are around 400 seconds long (7 minutes) and their naming conventions and brief description are shown in Table A.1.

Other characteristics of the generation and recording of the datasets are shown in Table A.2.

With these characteristic we can observe the quality of the datasets provided. The 20 Mhz bandwidth of the front-end filter allows for the use of techniques that search the correlation between C/A code and the P(Y) code in order to discriminate

Table A.1 TEXBAT datsets scenarios description, as given in [38]

Name	Description	Power Adv. [dB]
Clean Static	Clean Static	N/A
Clean Dynamic	Clean Dynamic	N/A
ds1	Static Switch	N/A
ds2	Static overpowered time push	10
ds3	Static matched-power time push	1.3
ds4	Static matched-power position push	0.4
ds5	Dynamic over-power time push	9.9
ds6	Dynamic matched-power position push	0.8
ds7	Static matched-power evolved time push	Variable

Table A.2 TEXBAT datsets characteristics

Name	Value
Sampling Frequency	25 Msps
Front-End Bandwidth	20 MHz
Bits of the ADC	16 bits
Intermediate Frequency	0 Hz
Samples format	I/Q interleaved

the spoofing attack. The 25 Msps and the 16 bit resolution provide a very good representation of the analog signal and the I/Q samples allow for coherent tracking of the signal [38]. The recording setup for the TEXBAT datasets is shown in Fig. A.1. We can observe how the spoofer is controlled from the computer in the bottom right corner. We see that the input of the spoofing device is the signal coming from the antenna, and the output of the spoofer is recombined with the antenna signal before being collected by the National Instruments hardware. In this way, they were able to generate a realistic intermediate spoofing attack through cable connection.

Two types of spoofing attacks were realized in the TEXBAT datasets. First one is denoted as a *time push* and it was done for scenarios ds2, ds3, ds5 and ds7. This type of attack, affects every satellite with the same code delay trend, creating a bias in all pseudoranges. This bias will result in an error in the time calculation of the PVT. This type of attack would be performed if the target of the spoofer is a synchronization system, e.g. a power grid or telecommunication cell. In Fig. A.2 we observe how the time component is affected for ds2 scenario where the blue plot is the spoofing trend, while the green one is the clean solution.

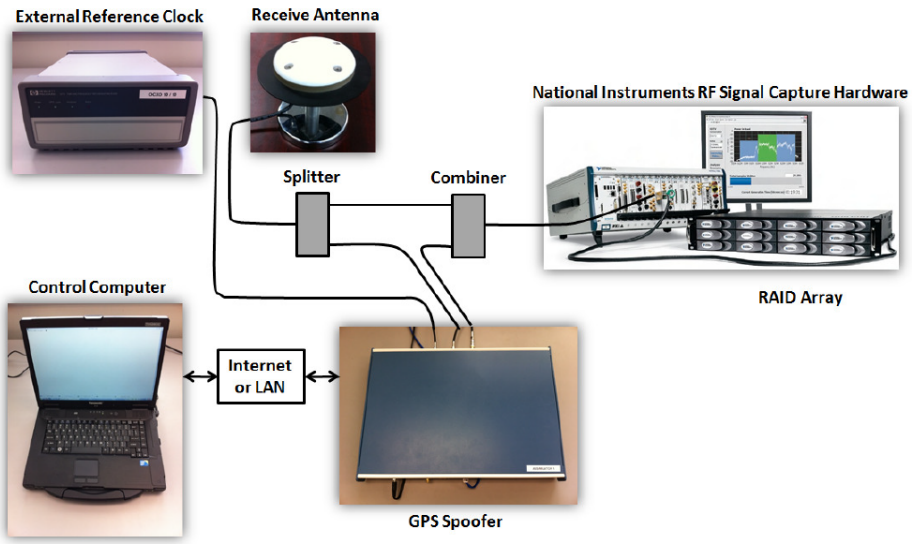

Fig. A.1 TEXBAT datasets recording setup. *Figure obtained from*[38]

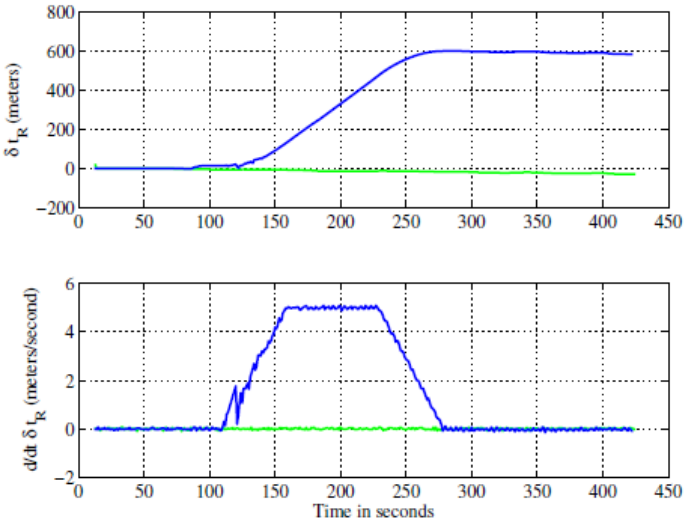

Fig. A.2 Time push attack for scenario ds2. We observe how in the top panel, the receiver clock offset δt is affected by the spoofing attack, shown in blue. In the bottom panel, the modifications to the clock offset rate $\dot{\delta t}$ are shown. The clean dataset results are shown in green. *From [38]*

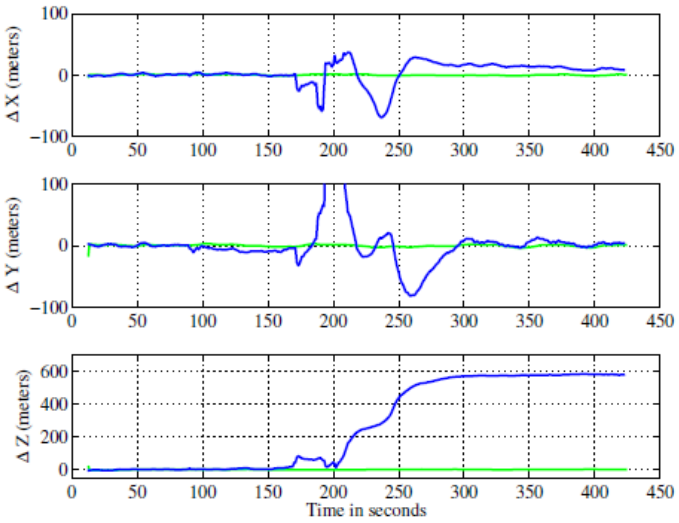

Fig. A.3 Position push attack for scenario ds4. The x, y and z trends denoting the errors in the coordinates are shown for both the clean (green) and spoofed (blue) scenarios. Errors up to 600 meters are introduced in the z component. *Figure obtained from*[38]

The second type of attack present in the TEXBAT is the *position push*. In this case, each satellite is affected by a different code delay, thus making the position solution of the PVT change according to the desires of the spoofing attack. These type of attacks are performed on scenarios ds4 and ds6. In Fig. A.3 we observe how the x, y and z solutions are affected by the spoofing attack. In blue the spoofing trends are shown, while in green the clean solution is plotted. We clearly observe how the main target in this case is the z component, and errors up to 600 meters are introduced on it.

Figures A.3 and A.2 provides a hindsight on the spoofing attacks' effects introduced in the TEXBAT and are useful to present some background knowledge for the analysis of results obtained throughout this thesis.

Overall, the TEXBAT provides a good collection of datasets to test anti-spoofing techniques, and it will be used with each of the detection algorithms analyzed in this thesis. External analysis of the datasets, including useful considerations for the TEXBAT processing were done in [50]. It is important to notice that, even if the collection presents different configurations of the spoofing, it is by no means a comprehensive set of all the different configurations that a spoofer can exploit.

Unfortunately, generating realistic datasets with spoofing attacks is not a simple task, and specialized hardware/software is needed. This situation makes the TEXBAT the best available solution for validating anti-spoofing algorithms and is the reason why it is used for the different techniques.

Static Scenarios

The static clean scenario was recorded with a rooftop antenna on the radionavigation laboratory building at the University of Texas at Austin. Four different spoofing scenarios are generated from the static collection:

- *Scenario ds1*. Is an antenna-switch scenario with the goal to simulate the scenario where a person with access to the antenna, switches the output of the antenna with the output of the spoofer. In this way the receiver is only able to *see* the signal coming from the spoofer, thus being fooled to follow its solution. This scenario is not used in this thesis because we are not interested in self-attacks, but we focus on external attackers.

- *Scenario ds2*. This scenario is an Overpowered time push, where the spoofer injects a signal with 10 dB power advantage over the satellite signal. This signal is aligned in code delay with the satellite signal. In this way the spoofer is able to gain control of the receiver easily and additional noise is added in order to maintain similar levels of C/N_0. With such a power advantage, asymmetries in the correlation function are neglected, but large amount of power is injected in the receiver front-end, detectable by means of the AGC gain. The time push means that only the time component of the receiver solution is affected once the attack is being performed.

- *Scenario ds3*. This scenario is similar to ds2, with the exception that the power advantage of the spoofer is only 1.3 dB. In this case, large asymmetries are revealed in the correlation function and the taking control of the receiver is not as smooth.

- *Scenario ds4*. Is the last scenario of the static set and it is very similar to ds3, with the exception that the position solution is modified instead of the time component and the power advantage is reduced to 0.4 dB. Thus being, every

satellite is affected by a different delay at a different time, according to the modifications that the spoofer wanted to do in the receiver position.

For static applications, we will focus our efforts in the datasets ds2, ds3 and ds4 since they simulate the effects of an external attacker affecting the receiver.

Dynamic Scenarios

Similar to what was done with the static scenario, a Dynamic Clean dataset was recorded in an urban scenario of Austin, TX. This dynamic set was recorded with an antenna on top of a van, driving around the city. Two dataset were generated using the dynamic scenario:

- *Scenario ds5*. This scenario is similar to ds2, where an overpowered time push is performed, but in this case the dynamic dataset is used as reference. The power advantage of the spoofer is also very close to the 10 dBs.
- *Scenario ds6*. This scenario is similar to ds4, where a spoofing signal with power advantage of only 0.8 dB with respect to the satellite signal is inserted, and a modification of the position solution is done.

Both dynamic datasets were used for testing road scenarios.

Update to the TEXBAT

In 2015, the first update of the TEXBAT was released with modifications to the way the spoofer alters the delay and the amplitude. Two new static scenarios were created and named ds7 and ds8 [35].

Scenario ds7, as explained in [35], is a static matched-power time push attack, based on the Clean Static dataset. In it, a spoofer signal is injected after 110 s, aligned in phase with the real signal, in the following 20 s the spoofing signal rotates its phase and amplitude until having an amplitude two times bigger w.r.t. the authentic signal amplitude and having a 180 deg rotation on the phase. In this way the spoofer slowly takes control of the receiver without the need to synchronize the navigation data bits [35]. The spoofing signal maintains its status for another 20 s where it

Table A.3 WAAS reference stations locations and duration of the datasets used in Chapter 5

Name	Location	Dur.for AGC	Dur. for SQM
HNL	Honolulu, HI	120 hrs.	24 hrs.
FAI	Fairbanks, AK	120 hrs.	24 hrs.
ZMA	Miami, FL	120 hrs.	24 hrs.
ZSE	Auburn, WA	120 hrs.	24 hrs.
ZBW	Nashua, NH	120 hrs.	24 hrs.
ZAU	Aurora, IL	120 hrs.	24 hrs.

could perform a navigation data bit attack and produces no relevant distortions on the correlation peak. At 150 s the spoofer starts the push off phase, linearly increasing the code delay at a rate of 1.2 meters per second and decreasing its amplitude until having the same amplitude than the original signal after 250 s.

The new type of attack is very dangerous given that the distortions created at the correlation functions at the beginning of the attack are relatively small and can prove difficult to detect via SQM techniques. Scenario ds8 is similar to ds7 but using an overpowered signal. Only ds7 was used during the thesis, since it is the matched power case that the techniques are more concerned about.

A.2 WAAS stations datasets using commercial off-the-shelf receivers

A Novatel G-III receiver was deployed in different WAAS stations in order to do several data-collections of clean and stable locations. These dataset can be used in order to obtain the baseline behavior for clean environments and to analyze the environmental situation of each of them. In Table A.3 the location of the WAAS stations used and the duration of each dataset is shown.

These datasets are used to asses the statistics of the AGC gain and the SQM metric, when no spoofer is present in Chapter 5. By means of these datasets we compute the thresholds used for spoofing detection algorithms. We refer to COTS receivers as the different GNSS receivers that are constructed by established and well known manufacturers, and are available for the public to buy. With this definition, we exclude all versions of software receivers and other receivers built ad-hoc in research labs for specific scientific purposes.

Table A.4 Dynamic urban scenarios used for assessment of multipath detection in Chapter 4

Name	Place	Date	Description
To-1	Turin, Italy	09/2013	Dynamic urban
To-2	Turin, Italy	02/2015	Dynamic urban

COTS receivers are used for many application, and provide different output messages containing the information required by the user. The messages written by a COTS receiver come in different fashions and each manufacturer usually has its own specific message format. As an example of an exchangeable message obtained from COTS receiver are the Receiver INdependent EXchange format (RINEX) files, developed originally by the Astronomical Institute of the University of Berne. RINEX files contain raw satellite navigation system data, such as pseudorange, carrier phase, Doppler shift and C/N_0 of each tracked satellite as well as the position, velocity and time solution computed from the receiver.

In our case, we use a Novatel G-III receiver, which is able to output a Novatel-specific message containing multicorrelator values and a message containing the values of the AGC gain. Using this receiver we collect multicorrelator measurements to build an SQM metric, that monitor the correlation function, and the AGC gain measurements, that is to be used as a power monitoring system.

A.3 Urban data collection from Turin, Italy

Several datasets were collected in urban environments in Turin, Italy. Two different types of datasets were collected:

- Dynamic scenarios in downtown Turin, where multipath signals are present due to the presence of buildings and narrow streets. The collection of the dynamic urban scenarios was performed by an antenna in the roof of a car, and collected using USRP devices connected to a computer. The car was driven around through different streets and configurations. An example of a trajectory done by the moving car is shown in Fig. A.4. The signals collected in this way are used in Chapter 4 for assessment of the metric β, in order to detect multipath signals. The naming convention along with the date of recording are shown in Table A.4.

Fig. A.4 Example of a dynamic urban data collection, obtained in downtown Turin, Italy. Position solutions are shown with red squares.

Table A.5 Static open sky scenarios used for building the confusion matrix in Chapter 7

Name	Place	Date	Description
static-1	Turin, Italy	09/2015	Static Rooftop
static-2	Turin, Italy	09/2015	Static Rooftop

- Static open sky scenario. These datasets were recorded using a rooftop antenna fixed on top of the ISMB building in Turin. These datasets provide a clean, static and open sky environment for assessment of the behaviors of the metrics in the absence of spoofing signals. They are used in Chapter 7 for the building of the confusion matrix, were no impairments are detected for these datasets. The brief details of the datasets is presented in Table A.5. In Fig. A.5 the PVT solution obtained by processing one of the datasets can be observed.

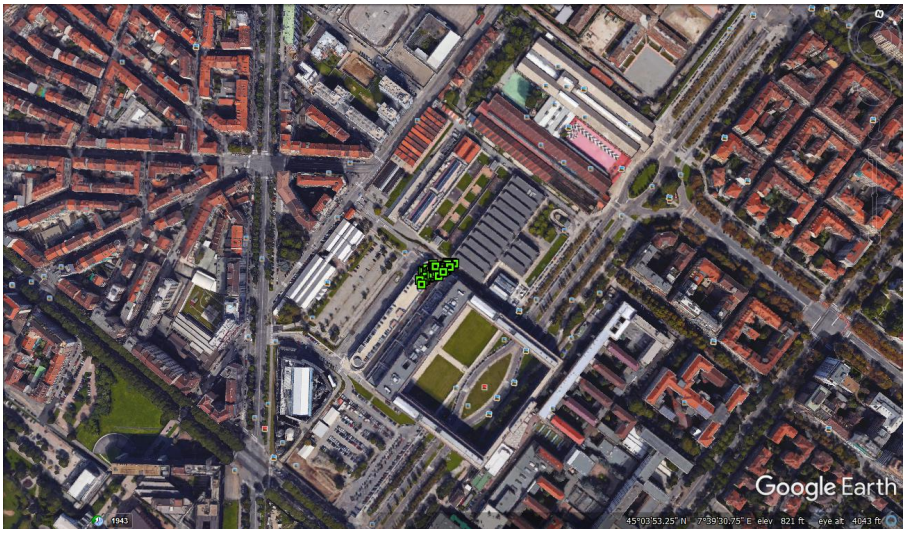

Fig. A.5 Example of a the static rooftop antenna data collection, obtained at ISMB in Turin, Italy. Positions are shown in green squares.

Appendix B

Threshold computation from Log-Likelihood ratio test

In this Appendix we write the equations for derivation of equation (3.8). We know that the LR is:

$$LR(M[k]) = \frac{p(M[k];H_1)}{p(M[k];H_0)} > \gamma_L \tag{B.1}$$

and under the hypothesis of high C/N_0 and $M \sim N(\mu, \sigma^2)$, we know that:

$$p(M[k];H_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(M[k]-\mu_i)^2}{2\sigma}} \tag{B.2}$$

so if we substitute (B.2) in (B.1), we obtain:

$$LR(M[k]) = \frac{\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(M[k]-\mu_1)^2}{2\sigma_1}}}{\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(M[k]-\mu_0)^2}{2\sigma_1}}} > \gamma_L \tag{B.3}$$

The threshold γ can then be derived as:

$$-(M[k]-\mu_1)^2 + (M[k]-\mu_0)^2 > 2\sigma_1 \ln \gamma_L \tag{B.4}$$

and

$$-M[k]^2 + 2M[k]\mu_1 - \mu_1^2 + M[k]^2 - 2M[k]\mu_0 + \mu_0^2 > 2\sigma_1 \ln \gamma \tag{B.5}$$

finally:

$$M[k] > \frac{\sigma_1 \ln \gamma_L}{\mu_1 - \mu_0} + \frac{\mu_1 + \mu_0}{2} = \gamma \tag{B.6}$$

That means that the final threshold γ, to which we need to compare the metric M against, is dependent on threshold γ_L that is unknown.

Practically, threshold γ can be directly computed from the desired false alarm probability of M, $P_{\mathrm{FA},M}$[46]. Under the hypothesis that $M \sim N(\mu, \sigma^2)$, the $P_{\mathrm{FA},M}$ is:

$$P_{\mathrm{FA,M}} = \frac{1}{2}\mathrm{erfc}\left(\frac{\gamma - \mu_{1,0}}{\sigma\sqrt{2}}\right) \tag{B.7}$$

thus:

$$\gamma = \sqrt{2}\sigma \cdot \mathrm{erfc}^{-1}(2P_{\mathrm{FA},M}) + \mu_0 \tag{B.8}$$

and it is only dependent on the distribution of M under hypothesis H_0.